TU Clausthal

Clausthal University of Technology

Modeling, Verification, and Strategic Reasoning in Multi-Agent Systems

Wojciech Jamroga

Modeling, Verification, and Strategic Reasoning in Multi-Agent Systems

Wojciech Jamroga

Cumulative habilitation at the Clausthal University of Technology, Germany

Contents

1 Introduction **8**
1.1 Logics for Imperfect Information Games 9
1.2 Plausible Behavior and Rational Play 11
1.3 Verification in Multi-Agent Systems 14
1.4 Strategic Reasoning for Stochastic Agent Systems 15

I Logics for Imperfect Information Games **18**

2 Constructive Knowledge **20**
2.1 Introduction 20
2.2 What Agents Can Achieve 24
2.2.1 ATL: Ability in Perfect Information Games 24
2.2.2 ATL with Epistemic Logic 27
2.2.3 Problems with ATEL 29
2.2.4 First Try: ATEL with Uniform Strategies 30
2.2.5 Aggregating Initial States: "Feasible Atel" 30
2.2.6 Going for Expressive Power: ATOL 31
2.2.7 Elegance and Simplicity: ATL_{ir} 32
2.2.8 Abilities of Rational Players: ETSL 33
2.2.9 Explicit Actions: ATEL-A 33
2.2.10 Other Possibilities 34
2.3 Constructive Strategic Logic 34
2.3.1 Language and Semantics 35
2.3.2 Additional Operators 37
2.3.3 Validity 39
2.4 Expressing Agents' Strategic Abilities 40
2.4.1 Capturing Levels of Strategic Power 40
2.4.2 Expressivity of CSL 42
2.4.3 Constructive Strategic Logic *vs*. ATEL 43
2.4.4 Constructive Strategic Logic *vs*. ETSL 45
2.5 Verification of Strategic Abilities through Model Checking . . 45
2.6 Constructive Knowledge 46
2.6.1 Properties of Constructive Knowledge 47
2.6.2 Is $\mathbb{K}_a$ an Epistemic Operator? 47
2.6.3 In Quest for the Truth Axiom 49
2.6.4 Properties of Collective Constructive Knowledge . . . 50
2.7 Negation, Localization, Definability of Knowledge 51

2.7.1 Local Evaluation of Formulae 51
2.7.2 Defining Standard Knowledge from Constructive Knowledge 52
2.7.3 Non-Standard Definitions of Negation 53
2.8 Normal Forms and Expressiveness 56
2.8.1 Constructive Normal Form 57
2.8.2 Expressiveness of Strong Negation 58
2.9 Conclusions 58
2.10 Appendix: Some Proofs 59

3 Playing Rationally vs. Knowing how to Play 68
3.1 Introduction 68
3.2 Reasoning about Abilities of Agents 69
3.2.1 ATL: Ability in Perfect Information Games 69
3.2.2 Strategic Ability and Incomplete Information 70
3.2.3 An Intuitive Semantics for Ability and Knowledge . . . 72
3.3 Epistemic Temporal Strategic Logic 73
3.3.1 The Semantics Made Easier to Read 74
3.3.2 A Few Properties 75
3.3.3 ETSL in Terms of Concurrent Epistemic Game Structures 77
3.4 Playing Rationally vs. Knowing how to Play 77
3.4.1 Rational Play of Individual Agents 78
3.4.2 Rational Coalitions Are at Disadvantage 80
3.5 Conclusions 81

II Plausible Behavior and Rational Play 84

4 Agents, Beliefs, and Plausible Behavior 86
4.1 Introduction 86
4.2 Branching Time and Knowledge 87
4.3 Extending Time and Knowledge with Plausibility 88
4.3.1 The Concept of Plausibility 89
4.3.2 CTLK with Plausibility 90
4.3.3 Defining Plausible Paths with Formulae 92
4.3.4 Comparison to Related Work 93
4.4 Plausibility, Knowledge, and Beliefs in CTLKP 96
4.4.1 Axiomatic Properties 96
4.4.2 Plausibility, Knowledge, and Beliefs 97
4.4.3 Properties of the Update 98
4.5 Verification of Plausibility, Time and Beliefs 99
4.6 Conclusions 100

5 Temporal Properties of Rational Play 102
5.1 Introduction 102
5.1.1 Idea and Main Results 103
5.1.2 Related Work 104
5.1.3 Structure of the Article 105
5.2 Preliminaries 106

5.2.1 Concepts From Game Theory . . . 106
5.2.2 ATL . . . 110
5.3 Relating Games and ATL-Like Logics . . . 111
5.3.1 ATL and Rational Play . . . 112
5.3.2 Game Logic with Preferences . . . 113
5.3.3 Models of ATL vs. Extensive Games . . . 113
5.3.4 ATLI and Solution Concepts . . . 115
5.3.5 General Solution Concepts . . . 118
5.4 The Logic ATLP . . . 119
5.4.1 The Language $\mathcal{L}^{\text{base}}_{ATLP}$ . . . 120
5.4.2 Semantics of $\mathcal{L}^{\text{base}}_{ATLP}$ . . . 121
5.4.3 Plausibility Terms based on ATLI . . . 125
5.4.4 Language $\mathcal{L}^{k}_{ATLP}$ and $\mathcal{L}^{\infty}_{ATLP}$ . . . 127
5.4.5 Semantics of $\mathcal{L}^{k}_{ATLP}$ and $\mathcal{L}^{\infty}_{ATLP}$ . . . 129
5.5 Properties of ATLP . . . 130
5.5.1 Embedding Existing Logics into ATLP . . . 130
5.5.2 Classical Solution Concepts in $\mathcal{L}^{1}_{ATLP}$ . . . 134
5.5.3 General Solution Concepts in $\mathcal{L}^{1}_{ATLP}$ . . . 136
5.6 Model Checking ATLP . . . 138
5.6.1 Model Checking $\mathcal{L}^{\text{base}}_{ATLP}$ . . . 140
5.6.2 Model Checking $\mathcal{L}_{ATLP^{ATLI}}$ . . . 145
5.6.3 Model Checking $\mathcal{L}^{k}_{ATLP}$ . . . 146
5.6.4 Summary of Complexity Results . . . 149
5.7 Conclusions . . . 151
5.8 Appendix: Bargaining with Discount . . . 152
5.9 Appendix: Uniform ATL_{ir} . . . 152
5.9.1 Semantics . . . 153
5.9.2 Model Checking Complexity . . . 153
5.10 Appendix: From ATL^{u}_{ir} to $ATLP^{ATLI}$ . . . 155
5.11 Appendix: Some Model Checking Complexity Proofs . . . 158
5.11.1 Results in Section 5.6.1 . . . 158
5.11.2 Results in Section 5.6.3 . . . 159

III Verification in Multi-Agent Systems 166

6 Model Checking Abilities of Agents 168
6.1 Introduction . . . 170
6.1.1 History of the Results . . . 171
6.2 ATL: A Logic of Strategic Ability . . . 172
6.2.1 Strategic Abilities with Concurrent Game Structures . . . 173
6.2.2 Semantics of ATL Based on ATS . . . 176
6.2.3 Beyond ATL: ATL^{+} and ATL^{*} . . . 177
6.2.4 Strategic Abilities under Imperfect Information . . . 178
6.3 Complexity of ATL Model Checking Revisited . . . 180
6.3.1 Model Checking ATL: Easy or Hard? . . . 181
6.3.2 "Positive ATL" Model Checking for CGS is $\mathbf{\Sigma_2^P}$-hard . . . 183
6.3.3 "Positive ATL" Model Checking for CGS is in $\mathbf{\Sigma_2^P}$ . . . 184
6.3.4 Full ATL . . . 186
6.4 Model Checking with Alternating Transition Systems . . . 186

6.4.1 Model Checking "Positive ATL" over ATS is in **NP** . . . 187
6.4.2 Single False Clause SAT . . . 188
6.4.3 Reduction of Sfc-SAT to "Positive ATL" Model Checking over ATS . . . 191
6.4.4 Full ATL . . . 192
6.4.5 Model Checking with Nondeterministic Transition Systems . . . 192
6.5 Turning Game Models Turn-Based . . . 194
6.5.1 Translation of Models . . . 195
6.5.2 Translation of Formulae . . . 196
6.6 Model Checking Agents: Incomplete Information . . . 198
6.6.1 Existing Results . . . 199
6.6.2 **NP**-completeness for "Positive ATL" . . . 201
6.6.3 Model Checking ATL_{ir} Is Indeed $\mathbf{\Delta}_2^{\mathrm{P}}$-complete . . . 203
6.6.4 The Complexity Refined . . . 206
6.6.5 Discussion . . . 209
6.7 Conclusions . . . 210

7 Modular Interpreted Systems 213
7.1 Introduction . . . 213
7.2 Logics of Time, Knowledge, and Strategic Ability . . . 214
7.2.1 Branching Time: CTL . . . 214
7.2.2 Adding Knowledge: CTLK . . . 215
7.2.3 Agents and Their Strategies: ATL . . . 215
7.2.4 Agents with Imperfect Information: ATL_{ir} . . . 216
7.3 Models and Model Checking . . . 217
7.3.1 Explicit Models . . . 217
7.3.2 Compressed Representations . . . 218
7.3.3 Interpreted Systems . . . 219
7.3.4 Concurrent Programs . . . 219
7.3.5 Synchronous CP and Simple Reactive Modules . . . 220
7.3.6 Concurrent Epistemic Programs . . . 221
7.3.7 Reactive Modules . . . 221
7.4 Modular Interpreted Systems . . . 222
7.4.1 Representing Agent Systems with MIS . . . 223
7.4.2 Modular Interpreted Systems vs. Simple Reactive Modules . . . 227
7.4.3 Model Checking Modular Interpreted Systems . . . 228
7.5 Conclusions . . . 229

IV Strategic Reasoning for Stochastic Multi-Agent Systems 230

8 A Temporal Logic for Markov Chains 232
8.1 Introduction . . . 232
8.1.1 Related Work . . . 233
8.2 Looking for Analogies . . . 234
8.2.1 Quantitative vs. Qualitative Models . . . 234
8.2.2 Quantitative vs. Qualitative Descriptions . . . 235

8.2.3 Logical operators as Minimizers and Maximizers 236
8.3 Markov Chains and Markov Decision Processes 236
8.3.1 Domain 236
8.3.2 Markov Chains 237
8.3.3 Markov Decision Processes 238
8.4 MTL_0: A Logic of Markov Chains 239
8.4.1 Syntax of MTL_0 240
8.4.2 Semantics of MTL_0 241
8.4.3 Levels of Truth 242
8.4.4 Transition Systems as Markov Chains. Correspondence between MTL_0 and CTL* 243
8.4.5 State-Based MTL_0 244
8.5 MTL_1: A Logic of Markov Decision Processes 245
8.5.1 Syntax and Semantics of MTL_1 245
8.5.2 Beyond MDP: the Multi-Agent Case 246
8.6 Comparison to DCTL 247
8.7 Conclusions 247

9 Temporal Logic for Stochastic MAS 249
9.1 Introduction 249
9.2 Markov Temporal Logic 250
9.2.1 Basic Models: Markov Chains 250
9.2.2 Logical Operators as Minimizers and Maximizers . . . 251
9.2.3 MTL_0: A Logic of Markov Chains 251
9.3 Reasoning about Stochastic Multi-Agent Processes 253
9.3.1 MTL_2: Syntax 253
9.3.2 MTL_2: Semantics 254
9.4 Formal Results 255
9.4.1 Levels of Truth 256
9.4.2 Concurrent Game Structures as MMDP's. Correspondence between MTL_2 and ATL* 256
9.4.3 State-Based Formulae and Bellman Equations 258
9.5 Conclusions 259

V Bibliography 262

Chapter 1

Introduction

Depending on the context, *multi-agent systems* [141, 144] are usually presented as a new paradigm for computation, programming, or design. Perhaps most importantly, they can be seen a philosophical metaphor that provides a way of modeling the world, and makes one use specific vocabulary when describing the phenomena we are interested in. Putting it in another way, multi-agent systems form a new paradigm for *thinking* and *talking* about the world, and assigning it a specific conceptual structure. Components of such systems are assumed to be autonomous, perhaps intelligent, definitely active or even pro-active... having some goals and beliefs... et cetera. Thus, the metaphor builds on the intuition that humans are agents, and that other entities we study can be just like us to some extent.

Logic-based methods for multi-agent systems have several advantages. Logic provides a vocabulary for naming properties of systems, and the vocabulary is given precise meaning via models and semantic rules. Moreover, logical models provide a conceptual apparatus for reasoning that can be as well used outside mathematical logic. In this thesis, we focus on modal logics with their clear and intuitively appealing conceptual machinery of *possible world semantics* (aka *Kripke semantics*). The logics draw from the long tradition of philosophical studies on human behavior and the behavior of the world in general, that yielded epistemic logic, deontic logic, temporal logic etc. In particular, we investigate a branch of modal logics that can be described as *strategic logics*. That is, the generic modal structure is infused with game-theoretical notions of *player*, *coalition* , *choice*, *strategy*, *outcome*, *rationality*, etc. Since all these concepts seem highly relevant for interaction between autonomous entities, it seems a perfect starting point for modeling and reasoning about multi-agent systems.

It should be pointed out that modal logics for multi-agent systems (and their models) can be used in at least two ways. First, we may strive to represent an objective observer's view to a multi-agent system with the instruments they provide. This is the viewpoint we usually adopt while talking about "specification", "design", "verification" etc. The observer (e.g., the designer or the administrator of the system) may collect all relevant aspects of the system in a Kripke model, and then derive or verify certain properties of this model. Or, the designer can specify some desirable properties of a system, and then try to engineer a model in which those properties hold. On

the other hand, a model can be also used to capture the *subjective* view of an agent to the reality he is acting in. In that case, the agent can ask about properties of the world via the properties of the model, or, more importantly, look for a strategy that makes some desirable property true in the model.

This thesis collects several papers that investigate multi-agent systems from different angles, using logics that combine concepts from game theory with well established modal treatment of time, knowledge and belief. Part I of the thesis represents the linguistic angle: we look for a language that allows to capture strategic abilities of agents under imperfect information in the most intuitive (and general) way. Part II is focused on the modeling dimension: we incorporate economic rationality in our view of agents, and study its impact on agents' behavior within the framework of modal logic. The perspective of Part III is computational: we study the complexity of verification for several specification languages, and observe some intriguing patterns among them. Finally, Part IV presents an attempt at integration of two fundamental approaches to reasoning about systems: we show how certain aspects of qualitative (logical) reasoning can be incorporated into quantitative analysis of MAS in the style of Markov decision processes.

1.1 Logics for Imperfect Information Games

Modal logics of strategic ability [6, 8, 113, 114] provide a language that successfully combines advantages of logic and game theory. The logics have clear and intuitive semantics, are axiomatizable, and have some interesting computational properties. Moreover, they are based on an intuitively appealing set of concepts that originate from logics of time and computation [116, 38, 44] and classical game theory [139, 109, 111]. Alternating-time temporal logic (ATL), proposed in [6] and further developed in [7, 8], is probably the most important logic of strategic ability that emerged in recent years. ATL is built around so called *cooperation modalities* $\langle\langle A\rangle\rangle$, where A is a coalition (i.e., a group of agents). Informally, $\langle\langle A\rangle\rangle\varphi$ says that group A has a joint strategy to ensure that, no matter what the other agents do, φ will become true.

From the game-theoretical perspective, ATL can be seen as a framework for reasoning about extensive form games of perfect information. The semantics of ATL addresses abilities of agents in these games in a convincing way,[1] but several ingredients are still missing if the logic is to be a good language for reasoning about behavior and interaction of agents. One such ingredient is *uncertainty*: real systems seldom include agents who always perfect information about the current state of the affairs. We study this issue in depth in Part I.

A number of logics have been proposed to capture agents' abilities in imperfect information scenarios [133, 134, 66, 83, 121, 86, 135, 3, 63], yet none of them seems the ultimate definitive solution. Most importantly, a coalition's ability to achieve property φ should imply that the agents have enough control and knowledge to *identify* and *execute* a strategy that enforces φ – and this turns out to be more sophisticated than it seems. In

[1] Although a debate on this subject has also begun recently, cf. [2, 21, 27, 140].

particular, in order to identify a successful strategy, the agents must consider not only the possible courses of action starting from the current state of the system, but also from states that the agents cannot distinguish from the current one. There are many subtle cases here, in which initial situations should be represented with different sets of states. The aim of the work reported in Chapter 2 is to come up with a logic that handles the subtleties in a way which is both general and elegant.

To achieve this, we build our proposal around new epistemic operators called *constructive knowledge* operators. The idea has been inspired by the tradition of constructivism which argues that one must present (or "construct") a mathematical object to prove that it exists [127]. In the same spirit, agents A *constructively know* that group B can enforce property φ if A can *present* a strategy for B that guarantees achieving φ. The logic which we propose has a fairly non-standard semantics: formulae are interpreted in *sets of states* rather than single states. Now, for a set of states Q, one can write $M, Q \models \langle\langle A \rangle\rangle \varphi$ to express the fact that A must have a strategy which is successful for all states in Q. Constructive knowledge operators yield sets of states for which a single evidence (i.e., a successful strategy) should be produced (instead of checking if the required property holds in each of the states separately, like standard epistemic operators do).

We call the resulting language *Constructive Strategic Logic* (CSL) to emphasize that, in order to prove $M, Q \models \varphi$ true, one must produce "constructive" evidence for all possible states in set Q, rather than "circumstantial" evidence that deals with every case $q \in Q$ separately.

In terms of technical results, we show in Chapter 2 that:

1. CSL is strictly more expressive than most existing solutions.

2. It retains the same model checking complexity, namely the model checking problem is $\mathbf{\Delta_2^P}$-complete with respect to the size of the model and the length of the formula.

3. Constructive knowledge satisfies axioms KD45. Moreover, a simple syntactical restriction is sufficient to guarantee validity of axiom T.

4. Standard knowledge can be defined as a special case of constructive knowledge.

We also present a number of examples to back up our claim that CSL is really the right language for qualitative reasoning about imperfect information scenarios in multi-agent systems.

Chapter 3 focuses on the relationship between CSL and one of the more interesting previous proposals, called Epistemic Temporal Strategic Logic (ETSL) [135]. Technically, this can be seen as a minor contribution that expands the picture from Chapter 2. However, we also propose an intuitive reading of the central operator in ETSL: "if A play *rationally* to achieve φ (meaning: they never play a dominated strategy), they will achieve φ". Thus, one may treat ETSL as a logic that captures properties of agents' rational play (albeit in a limited sense). Under this interpretation, we use CSL and ETSL to prove that a rational player knows that he will succeed if, and

only if, he knows how to play to succeed. Moreover, we show that the same is not true for rational coalitions of players.

Among other things, the results in Chapter 3 touch upon the other important ingredient which is missing in the original ATL, namely constraints on agents' behavior that stem from their rationality. This connects nicely to the subject of Part II, where we discuss how rationality can be modeled and specified, and how one can reason about the outcome of rational play.

Part I of the thesis is based on the following papers:

1. Wojciech Jamroga and Thomas Ågotnes (2007), Constructive Knowledge: What Agents Can Achieve under Imperfect Information. Journal of Applied Non-Classical Logics, 17(4), pp. 423–475.

2. Wojciech Jamroga (2006), On the Relationship between Playing Rationally and Knowing how to Play: A Logical Account. Proceedings of the 29th German Conference on Artificial Intelligence KI'06, pp. 403–417, Lecture Notes in Computer Science.

1.2 Plausible Behavior and Rational Play

Logics of knowledge and belief are often too static and inflexible to be used on real-world problems. In particular, they usually offer no concept for expressing that some course of events is *more plausible* than another. This, and especially the notion of plausibility provided by game-theoretical solution concepts, is the main focus of Part II. We propose how plausibility and rationality can be modeled, and how one can reason about the outcome of plausible behavior and/or rational play on top of these models.

Chapter 4 addresses the general perspective. We extend modal logics of time and knowledge with a notion of *plausible behavior*: the notion is added to the language of CTLK [115], which is a straightforward combination of the branching-time temporal logic CTL [41, 38] and standard epistemic logic [57, 42]. In our approach, plausibility is modeled as a temporal property of behaviors. That is, some behaviors of the system can be assumed plausible and others implausible, with the underlying idea that the latter should perhaps be ignored in practical reasoning about the future. Behaviors are formally modeled as computations (or paths) in the system. Since we use branching-time logic as the starting point, the resulting language allows for reasoning about what can (or must) plausibly happen.

Moreover, we propose a particular notion of beliefs (inspired by [126, 46]), defined on top of knowledge and plausibility. The main intuition is that beliefs are facts *that an agent would know if he assumed that only plausible things could happen*. An actual notion of plausibility can emerge in many different ways. For example, it may result from observations and learning, knowledge exchange, folk knowledge, or adopting a rationality criterion from game theory.

We believe that humans use such a concept of plausibility and "practical beliefs" quite often in their everyday reasoning. Restricting one's reasoning to plausible possibilities is essential to make the reasoning feasible, as the space of *all* possibilities is exceedingly large in real life. Of course, this does

not exclude a more extensive analysis in special cases, e.g. when our plausibility assumptions do not seem accurate any more, or when the cost of inaccurate assumptions can be too high (as in the case of high-budget business decisions). But even in these cases, we usually do not get rid of plausibility assumptions completely – we only revise them to make them more cautious. That is, when planning to open an industrial plant in the UK, we will probably consider the possibility of our main contractor taking her life, but we will still *not* take into account the possibilities of: an invasion of UFO, England being destroyed by a meteorite, Fidel Castro becoming the British Prime Minister etc. Note that this is fundamentally different from using a probabilistic model in which all these unlikely scenarios are assigned very low probabilities: in that case, they also have a very small influence on our final decision, but we must process the *whole* space of physical possibilities to evaluate the options.

We investigate some important properties of plausibility, knowledge, and belief in this new framework. In particular, we show that:

1. Knowledge is an **S5** modality (as expected), and that beliefs satisfy axioms **K45** in general, and **KD45** for the class of *plausibly serial models*.

2. The relationship between knowledge and belief for plausibly serial models is natural and reflects the initial intuition well.

3. Plausibility assumptions can be specified in the object language through a *plausibility update operator*, with natural properties.

4. Model checking formulae of CTLK with plasibility is no more complex than model checking CTL and CTLK, i.e., it is **P**-complete in the size of the model and the length of the formula.

Several modal notions of plausibility were already discussed in [46, 47, 126, 107, 93]. In these papers, like in ours, plausibility is used as a primitive semantic concept that helps to define beliefs on top of agents' knowledge. However, *our* plausibility is defined as a temporal property, i.e., it is a property of temporal paths rather than states, and we use branching (rather than linear) time with explicit quantification over temporal paths. Moreover, our framework is more computationally oriented since its semantics is based on concepts and representations that were devised to model interaction of processes in computational systems. Finally, our logic provides a mechanism for specifying and updating sets of plausible paths in the object language. Thus, plausibility sets can be specified in a succinct way, which is another feature that makes our framework computation-friendly. The model checking results are especially encouraging in this light.

All these differences are rather subtle. Chapter 5 departs more significantly from the existing modal approaches to plausible reasoning. Here, the notion of plausibility is given a game-theoretical foundation. Game theory identifies a number of *solution concepts* (e.g., Nash equilibrium, undominated strategies, Pareto optimality) that can be used to define rationality of players. Then, one can *assume that players play rationally* in the sense of one of the concepts, and we *ask about the outcome of the game under this assumption*. Note that solution concepts do not only help to determine the right decision for "our" agent. Perhaps even more importantly, they constrain

the possible (predicted) responses of the opponents. For many games the number of all possible outcomes is infinite, although only some of them "make sense". Still, we need a notion of rationality (like subgame-perfect Nash equilibrium) to discard the "less sensible" ones, and determine what should happen had the game been played by ideal players.

There are two possible points of focus in this context. Research within game theory understandably favors work on *characterization* of various types of rationality (and defining most appropriate solution concepts). Applications of game theory, also understandably, tend toward *using* the solution concepts in order to predict the outcome in a given game (in other words, to "solve" the game). The first issue has been studied in the framework of logic, for example in [11, 19, 124, 125]; more recently, game-theoretical solution concepts have been characterized in dynamic logic [62, 61], dynamic epistemic logic [14, 128], and ATL [130, 85]. The second issue seems to have been neglected in logic-based research: papers by Van Otterloo and his colleagues [137, 138, 136, 135] are the only exceptions we know of (and each of them commits to a particular view of rationality). Here, we try to fill in this gap, and propose a general, modal logic-based framework for reasoning about behavior and abilities of rational agents. We also attempt to show that, with temporal formulae, much more can be said about the outcome of rational play than just the payoff values at the end of the game.

Chapter 5 includes the following results:

1. We extend ATL to a new logic ATLP ("ATL with Plausibility") that allows to reason about what agents can achieve under an arbitrary plausibility assumption. E.g., one can assume that the agents can only play Nash equilibria, or undominated strategies, and ask which outcomes can be obtained by whom under this assumption.

2. We extend the results from [130, 85], and show that the classical solution concepts (*Nash equilibrium*, *subgame perfect Nash equilibrium*, *Pareto optimality*, and others) can be characterized in ATLP. In consequence, ATLP can serve both as a language for reasoning about rational play, and for specifying what rational play is.

3. We propose an alternative approach to defining solution concepts for infinite games by specifying "winning conditions" with temporal formulae. We also demonstrate how these "qualitative" solution concepts (parameterized with temporal formulae) can be characterized in ATLP.

4. We constructively show that several relevant logics can be embedded into ATLP.

5. We show that, for different subclasses of the new logic, the complexity of model checking ATLP ranges from $\mathbf{\Delta_3^P}$-completeness to **PSPACE**-completeness. We also argue that, when the number of plausible strategy profiles is reasonably small, the model checking can be done in polynomial time.

Part II of the thesis is based on the following papers:

1. Nils Bulling and Wojciech Jamroga (2007), Agents, Beliefs, and Plausible Behavior in a Temporal Setting. Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems AAMAS'07, pp. 570–577.

2. Nils Bulling, Wojciech Jamroga, and Jürgen Dix (2008), Reasoning about Rational Agents in ATLP. Annals of Mathematics and AI. To appear.

1.3 Verification in Multi-Agent Systems

A study of computational complexity is nowadays almost obligatory in a paper on logic in MAS. Authors usually study the complexity of model checking and/or satisfiability checking of their logic in order to back the usefulness of the proposal with a formal argument. Unfortunately, the results bear often only a limited connection to the "practical" complexity which is encountered when one tries to use the formalism in reality. Moreover, it is possible to manipulate the context so that different complexity results are obtained for the same problem. In Part III, we study the model checking complexity for some variants od ATL with perfect and imperfect information. In particular, we show that verification with ATL is not as cheap as it seems according to the complexity results from [7, 8]. Moreover, we demonstrate how the complexity class of the model checking problem changes when we change the way we represent input and/or measure its size.

1. It is well known that ATL model checking can be done in time linear in the size of the model. In Chapter 6, we point out that the size of an ATL model is usually *exponential in the number of agents*. Moreover, when the size of models is defined in terms of *states and agents* rather than *transitions*, it turns out that the problem is $\mathbf{\Delta_3^P}$-complete for concurrent game structures, and $\mathbf{\Delta_2^P}$-complete for another class of ATL models called alternating transition systems.

2. We show that for "Positive ATL" that allows for negation only on the level of propositions, model checking is somewhat cheaper, namely it is $\mathbf{\Sigma_2^P}$-complete for concurrent game structures, and $\mathbf{NP}$-complete for alternating transition systems.

3. We present a nondeterministic polynomial reduction from checking arbitrary alternating transition systems to checking turn-based transition systems. We also discuss the determinism assumption in alternating transition systems, and show that it can be easily removed.

4. We study the model checking complexity for a variant of ATL with imperfect information (ATL_{ir}). We show that the problem is $\mathbf{\Delta_2^P}$-complete in the number of transitions and the length of the formula, therefore closing a gap in previous work of Schobbens [121]. Furthermore, we prove that model checking ATL_{ir} is $\mathbf{\Delta_3^P}$-complete with respect to the number of agents, states and actions in the concurrent game structure ($\mathbf{\Sigma_2^P}$-complete for "Positive ATL_{ir}"). Thus, model checking of agents'

abilities for imperfect information appears harder than for perfect information when a coarser measure of the input is used, but both problems fall into the same complexity class when a finer-grained measure is used.

5. Finally, in Chapter 7, we propose a new class of representations that can be used for modeling (and model checking) temporal, strategic and epistemic properties of agents and their teams. Our representations follow the main ideas from *interpreted systems* of Halpern, Fagin et al. [58, 42]; however, they are also modular and compact in the way *concurrent programs* [92] are. We conclude with a somewhat surprising result that model checking ability under imperfect information for the new class can be computationally cheaper than model checking perfect information.

The resulting pattern of complexity is at least intriguing. Perfect information makes model checking cheaper, comparable, or harder than imperfect information, depending on the abstraction level and the way we define the size of the input. Does it mean that theoretical complexity results are not worth anything in practice? Not necessarily – but certainly one needs to take these results with a grain of salt. In most cases, only a more extensive study (carried out from several different perspectives) can give us a meaningful picture of the *real* computational difficulty behind the problem.

This part of the thesis is based on the following papers:

1. Wojciech Jamroga and Jürgen Dix (2008), Model Checking Abilities of Agents: A Closer Look. Theory of Computing Systems, 42(3), pp. 366–410.

2. Wojciech Jamroga and Thomas Ågotnes (2007), Modular Interpreted Systems. Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems AAMAS'07, pp. 892–899.

1.4 Strategic Reasoning for Stochastic Agent Systems

Models of multi-agent systems can be, roughly speaking, divided into two classes. *Qualitative models* provide no numerical information about states of the system and relationships between states. They are used as basic models of computation, in semantics of programming languages, and in specification and verification of systems. They are also used in domains where quantitative information cannot be reliably obtained nor assumed. *Quantitative models*, on the other hand, assume that relationships are measurable, and provide numerical information about the degrees of relations. For the relations between states, the degrees are usually given as probabilities. For the "qualities" of particular states, one often talks about *rewards* or *utilities*. Quantitative representations are used in stochastic modeling, decision theory, reinforcement learning, game theory etc.

In the previous chapters, we were concerned with analysis of agent systems, based on qualitative models and descriptions. Part IV presents an attempt to transfer some intuitions from qualitative approaches to the quantitative realm. That is, we explore analogies between transition systems and Markovian models in order to provide a more expressive language for reasoning about, and specification of agents in stochastic environments.

Quantitative analysis of processes is usually based on the notion of expected reward. Still, other features of Markov chains and Markov decision processes can be also interesting. In Chapter 8, we propose to use the methodology of propositional modal logic in order to study quantitative properties of systems and processes. We observe that – when truth values represent utility of an agent – temporal operators "sometime" and "always" have a very natural interpretation. "Sometime p" ($\Diamond p$) can be rephrased as "p is achievable in the future". Thus, under the assumption that agents want to obtain as much utility as possible, it is natural to view the operator as maximizing the utility value along a given temporal path. Similarly, "always p" ($\Box p$) can be rephrased as "p is guaranteed from now on". In other words, $\Box p$ asks for the minimal value of p on the path. On a more general level, every universal quantifier is essentially a minimizer of truth values, while existential quantifiers can be seen as maximizers. Thus, $\mathsf{E}\gamma$ ("there is a path such that γ") maximizes the utility specified by γ across all paths that can occur; likewise, $\mathsf{A}\gamma$ ("for all paths γ") minimizes the value of γ across paths. Also, disjunction and conjunction can be seen as a maximizer and a minimizer: $\varphi \vee \psi$ reads easily as "the utility that can be achieved through φ or ψ", while $\varphi \wedge \psi$ reads as "utility guaranteed by both φ and ψ".

Markov temporal logic (MTL), proposed in Chapter 8, makes use of the above intuitions. We also briefly consider an extension for Markov decision processes where a single decision maker influences the behavior of the system by choosing a stochastic policy. In terms of technical results, we show that:

1. The simplest version of MTL (for Markov chains) strictly extends the branching-time logic CTL*,

2. Typical fixpoint properties for temporal operators do also hold for the "state-based" subset of MTL.

Related work includes research on multi-valued modal logics [50, 36, 90, 97] and probabilistic model checking [32, 65, 12]; our work comes especially close to [33], where the "Discounted CTL" was proposed. However, we attempt at a more *systematic* exploration of linguistic features that are offered by propositional modal logic. The benefits of this approach can be already seen in Chapter 9, where Markov temporal logic is extended to the multi-agent case in a straightforward way. We show that the resulting logic strictly extends $\mathrm{ATL}_{\mathrm{Ir}}$*, i.e., alternating-time temporal logic with memoryless strategies. We also present fixpoint characterizations for some natural combinations of strategic, path, and temporal operators, that can be seen as analogues of Bellman equation. The characterizations enable computing the truth values of many MTL formulae by solving sets of simple equations.

This part of the thesis is based on the following papers:

1. Wojciech Jamroga (2008), A Temporal Logic for Markov Chains. Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems AAMAS'08, pp. 697–704.

2. Wojciech Jamroga (2008), A Temporal Logic for Stochastic Multi-Agent Systems. Proceedings of the 11th Pacific Rim International Conference on Multi-Agents PRIMA'08, Lecture Notes in Computer Science, pp. 239–250. To appear.

Part I

Logics for Imperfect Information Games

Chapter 2

Constructive Knowledge: What Agents Can Achieve Under Imperfect Information (joint work with Thomas Ågotnes)

Abstract. We propose a non-standard interpretation of Alternating-time Temporal Logic with imperfect information, for which no commonly accepted semantics has been proposed yet. Rather than changing the semantic structures, we generalize the usual interpretation of formulae in *single* states to *sets* of states. We also propose a new epistemic operator for "practical" or "constructive" knowledge, and we show that the new logic (which we call Constructive Strategic Logic) is strictly more expressive than most existing solutions, while it retains the same model checking complexity. Finally, we study properties of constructive knowledge and other operators in this non-standard semantics.

`Keywords:` Alternating-time Temporal Logic, strategic ability, imperfect information, epistemic logic.

2.1 Introduction

Modal logics of strategic ability [6, 8, 113, 114] form one of the fields where logic and game theory can successfully meet. The logics have clear possible worlds semantics, are axiomatizable, and have some interesting computational properties. Moreover, they are underpinned by a clear and intuitively appealing conceptual machinery for modeling and reasoning about systems that involve multiple autonomous agents. The basic notions, used here, originate from temporal logic (i.e., the logic of time and computation) [116, 38, 44], and classical game theory [139, 109, 111] which emerged in an at-

tempt to give precise meaning to common-sense notions like choices, strategies, or rationality – and to provide formal models of interaction between autonomous entities. Modal logics that embody basic game theory notions – and at the same time build upon branching-time temporal logics, well known and studied in the context of computational systems – seem a good starting point for investigating multi-agent systems.

Alternating-time Temporal Logic (ATL), proposed in [6] and further developed in [7, 8], is probably the most important logic of strategic ability that has emerged in recent years. The key elements of ATL are so called *cooperation modalities* $\langle\langle A \rangle\rangle$, one for each possible set of agents A. Informally, the meaning of $\langle\langle A \rangle\rangle \varphi$ is that the group A has a joint strategy to ensure that, no matter what the other agents do, φ will become true. However, ATL considers only agents that possess perfect information about the current state of the world, and such agents seldom exist in reality. On the other hand, imperfect information and knowledge are addressed in epistemic logic in a natural way [57]. A combination of ATL and epistemic logic, called *Alternating-time Temporal Epistemic Logic* (ATEL), was introduced in [133, 134] to enable reasoning about agents acting under imperfect information. Still, it has been pointed out in several places [66, 83, 86, 1] that the meaning of ATEL formulae can be counterintuitive. Most importantly, an agent's ability to achieve property φ should imply that the agent has enough control and knowledge to *identify* and *execute* a strategy that enforces φ.

Example 1 *Let us consider a variant of the example from [121]. There is a banker* b *(who knows the code that opens the safe), and a robber* r *who does not know the code. The banker can also change the code, and he does so from time to time. If a person is in the vault, and types the code correctly, the safe opens. If incorrect code is typed, the vault door closes, jailing the person inside.*

Intuitively, there is no feasible plan for r *to quickly open the safe whenever he wants to (unless he threatens or corrupts the banker to reveal the code). Reason: whatever the current code is, the vault looks the same to* r*, and a sensible plan should specify the same choices in indistinguishable situations (otherwise the plan cannot be executed). On the other hand, there* is *a behavior specification (formally: a function from states to actions) that allows* r *to rob the bank, and it reads as follows: "if you are outside then enter the vault; if you are inside and the code is* 00000 *then type* 00000*; if the code is* 00001 *then type* 00001 *etc.". Clearly, not every specification like this makes up a strategy that can be executed by the player. Those that do are sometimes called* uniform strategies*, and are required to prescribe the same choices in indistinguishable states.*[1] *Unfortunately, ATEL accepts all functions from states to actions as a strategies, which does not blend well with the assumption that agents' knowledge is limited.*

It should be noted that it is not always enough to restrict strategies to uniform ones. Consider a situation when b *has set the code to* 23087 *and gone for lunch (so he will not change it again for a while), and* r *is now standing in front of the safe. Obviously, there is a uniform strategy for* r *that leads to opening the safe, namely: "type* 23087*, regardless of anything". The robber even knows that such a successful strategy exists. On the other hand, he does not know which strategy it is (because*

[1] This very much in agreement with game-theoretical treatment of games with imperfect information. A strategy in such games is a function from *information sets* (i.e., sets of indistinguishable states) to actions.

he does not know what the current state is), and thus he does not have the ability to open the safe for sure.

Reasoning about the collective abilities of teams requires even more sophisticated concepts.

Example 2 *Suppose that, instead of a single robber r, a gang of robbers $r_1, ..., r_n$ is operating. If they can discuss their plans before acting, they can share their individual information about the current state of affairs in order to determine the best strategy (which seems to somehow be related to the notion of* distributed knowledge *from epistemic logic). If they have to coordinate on the fly, without communicating, then it is desirable that they all can separately identify the same winning strategy, and they all know that the others can identify this strategy, and they all know that they all know etc. (which looks very much like* common knowledge*). Thus, there seems to be no single notion of collective knowledge that suffices for all possible scenarios involving collective strategic ability.*

Example 3 *Let us also consider an industrial company that wants to start production, and looks for a good strategy when and how it should do it. Such a strategy is feasible if it can be carried out by the company (i.e., by its management and employees). However, it does not have to be* prepared *by members of the company themselves. In many cases, a consulting firm is hired to work out the best plan. Then, it is enough that members of the consulting firm can work out a good strategy which can be executed by the management and employees of the industrial company.*

A number of logics were proposed to capture these, and similar, properties [66, 83, 121, 86, 135, 63], yet none of them seems the ultimate definitive solution. Most of the solutions agree that only uniform strategies should be taken into account (cf. Example 1). However, in order to identify a successful strategy, the agents must consider not only the possible courses of action starting from the current (actual) state of the system, but also from states that the agents cannot distinguish from the current one. There are many variants here, especially when group epistemics is concerned, as Examples 2 and 3 demonstrate. The agents may have common, mutual, or distributed knowledge[2] about a strategy being successful, or they may be hinted the right strategy by a distinguished member (the "boss"), a subgroup ("headquarters committee") or even another group of agents ("consulting company") etc. In other words, there are many subtle cases in which the (subjectively possible) initial situations should be represented with different sets of states. Some existing solutions treat only some of the cases (albeit often in an elegant way), while the others offer a very general treatment of the problem at the expense of an overblown logical language (which is by no means elegant). *Our aim is to come up with a logic of ability under imperfect information, which is both general and elegant.* By "general", we mean that it allows to characterize as many meaningful levels of strategic ability as possible (and at least as many as ATOL [83]). In particular, it should enable the distinction between various readings of knowing a strategy "de re" and "de dicto" for individual as well as collective players. By "elegant", we mean that it allows

[2] See Section 2.2.2 for precise definitions.

us to express various levels of ability by *composition* of epistemic operators with strategic operators, instead of assigning a specialized modality to every conceivable combination.

To achieve this, we build our proposal around new epistemic operators for what we call "practical" or "constructive" knowledge. The idea has been inspired by the tradition of *constructivism* which argues that one must find (or "construct") a mathematical object to prove that it exists [127]. In the same spirit, agents A *constructively know* that $\langle\langle B\rangle\rangle\varphi$ if they can present a strategy for B that guarantees achieving φ. The logic which we propose in this paper has a fairly non-standard semantic interpretation. We use the same semantic structures that were used before for ATEL, ATOL, ATL$_{ir}$ etc.; however, in our semantics formulae are interpreted over *sets of states* rather than single states. This reflects the intuition that the "constructive" ability to enforce φ means that the agents in question have a single strategy that brings about φ for *all* subjectively possible initial situations – and not merely that a successful strategy exists for *each* initial situation (because those could be different strategies for different situations). To do it in a flexible and general way, the type of satisfaction relation in our proposal forces one to specify the set of initial states explicitly. In consequence, we write $M, Q \models \langle\langle A\rangle\rangle\varphi$ to express the fact that A must have a strategy which is successful for all states in a set of states Q.

Semantically, the constructive knowledge operators yield sets of states for which a single evidence (i.e., a successful strategy) should be presented (instead of checking if the required property holds in each of the states separately, like standard epistemic operators do). For example, $M, q \models \mathbb{K}_a\langle\langle a\rangle\rangle\varphi$ holds iff $\langle\langle a\rangle\rangle\varphi$ is satisfied by M, Q, where Q is the set of states which agent a cannot distinguish from q. We point out that the new operators capture the notion of knowing "de re", while the standard epistemic operators refer to knowing "de dicto". We call the resulting logic *Constructive Strategic Logic* (CSL) to emphasize that, in order to prove $M, Q \models \varphi$ true, one must produce "constructive" evidence for all possible cases in Q, rather than "circumstantial" evidence that deals with every case $q \in Q$ separately.

We begin with a short presentation of Alternating-time Temporal Logic and the attempts that have been made to extend ATL to scenarios with imperfect information (Section 2.2). In Section 2.3 we present the main contribution of this paper: a new, non-standard semantics for the logic of strategic ability, imperfect information and knowledge. We show that it is strictly more expressive than the existing solutions, with the possible exception of ETSL (Section 2.4), while it retains the same model checking complexity (Section 2.5). Then, in Section 2.6, we study the properties of constructive knowledge itself. It turns out that, when "standard" knowledge is assumed to be S5, constructive knowledge is KD45. Moreover, a simple syntactical restriction is sufficient to guarantee validity of axiom T for constructive knowledge. In Section 2.7 we show that standard knowledge is definable from constructive knowledge. We also observe that, when we allow a formula to be interpreted in a set of states, several definitions of negation (corresponding to different ways of quantifying over the set) are possible. We introduce and discuss such alternative negations and related operators. Finally, in Section 2.8 we investigate the relative expressiveness of some of these operators in detail, and we define a normal form for formulae of our

language.

Some preliminary results of this research have been reported in [73, 75].

2.2 What Agents Can Achieve

Alternating-time Temporal Logic ATL [6, 7, 8] was introduced by Alur, Henzinger and Kupferman in order to capture properties of *open computational systems* (such as computer networks), where different components can act autonomously. Computations in such systems are effected by the components' combined actions. Alternatively, ATL can be seen as a logic for systems involving multiple agents, that allows one to reason about what agents can achieve in game-like scenarios. As ATL does not include imperfect information in its scope, it can be seen as a logic for reasoning about agents who always have complete knowledge about the current state of affairs.

2.2.1 ATL: Ability in Perfect Information Games

ATL can be understood as a generalization of the branching time temporal logic CTL [29, 38], in which path quantifiers are replaced with so called *cooperation modalities*. The formula $\langle\langle A\rangle\rangle\varphi$, where A is a coalition of agents, expresses that A have a collective strategy to enforce φ. ATL formulae include temporal operators: "$\bigcirc$" ("in the next state"), $\Box$ ("always from now on") and $\mathcal{U}$ ("until"). Operator $\Diamond$ ("now or sometime in the future") can be defined as $\Diamond\varphi \equiv \top\,\mathcal{U}\,\varphi$. Similarly to CTL, every occurrence of a temporal operator is immediately preceded by exactly one cooperation modality.[3] The broader language of ATL*, in which no such restriction is imposed, is not discussed in this paper.

Formally, the recursive definition of ATL formulae is:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle\bigcirc\varphi \mid \langle\langle A\rangle\rangle\Box\varphi \mid \langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\varphi$$

where A is a set of agents.

Example ATL properties are: $\langle\langle jamesbond\rangle\rangle\Diamond\mathsf{win}$ (James Bond has an infallible plan to eventually win), and $\langle\langle jamesbond, bondsgirl\rangle\rangle\mathsf{fun}\,\mathcal{U}\,\mathsf{shot}-\mathsf{at}$ (Bond and his current girlfriend have a collective way of having fun until someone shoots at them).

A number of semantics have been defined for ATL, most of them equivalent [51, 52]. In this paper, we use a variant of *concurrent game structures* (CGS s) as models. A CGS is a tuple $M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o\rangle$ which includes a nonempty finite set of all agents $\mathbb{A}\mathrm{gt} = \{1, ..., k\}$, a nonempty set of states St, a set of atomic propositions Π, a valuation of propositions $\pi : St \to 2^{\Pi}$, and a set of (atomic) actions Act. Function $d : \mathbb{A}\mathrm{gt} \times St \to (2^{Act} \setminus \emptyset)$ defines nonempty sets of actions available to agents at each state, and o is a (deterministic) transition function that assigns the outcome state $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to state q and a tuple of actions $\langle\alpha_1, \ldots, \alpha_k\rangle$, $\alpha_i \in d(i, q)$, that can be executed by $\mathbb{A}\mathrm{gt}$ in q. A *strategy* s_a of agent a is a conditional plan that specifies what a is going to do for every possible situation: $s_a : St \to Act$

[3] The logic to which such a syntactic restriction applies is sometimes called *"vanilla"* ATL (resp. "vanilla" CTL etc.).

such that $s_a(q) \in d(a,q)$. A *collective strategy* S_A for a group of agents A is a tuple of strategies, one per agent from A.

Remark 1 *This is a deviation from the original semantics of ATL [6, 7, 8], where strategies assign agents' choices to* sequences *of states, which suggests that agents can by definition recall the whole history of each game. Both types of strategies yield equivalent semantics for "vanilla" ATL, but the choice of one or the other notion of strategy does affect the semantics of the full ATL* and most ATL variants for games with imperfect information [121]. The main reason why we use "memoryless" strategies here is that model checking strategic abilities of agents with perfect recall and imperfect information is believed to be undecidable (cf. Section 2.2.10).*

A *path* λ in model M is an infinite sequence of states that can be effected by subsequent transitions, and refers to a possible course of action (or a possible computation) that may occur in the system; by $\lambda[i]$, we denote the ith position on path λ. Function $out(q, S_A)$ returns the set of all paths that may result from agents A executing strategy S_A from state q onward:

$out(q, S_A) = \{\lambda = q_0 q_1 q_2 ... \mid q_0 = q$ and for every $i = 1, 2, ...$ there exists a tuple of agents' decisions $\langle \alpha_1, ..., \alpha_k \rangle$ such that $\alpha_a = S_A(a)(q_{i-1})$ for each $a \in A$, and $\alpha_a \in d(a, q_{i-1})$ for each $a \notin A$, and $o(q_{i-1}, \alpha_1, ..., \alpha_k) = q_i\}$.

Informally speaking, $M, q \models \langle\langle A \rangle\rangle \varphi$ iff there is a collective strategy S_A such that φ holds for every $\lambda \in out(q, S_A)$. Formally, the semantics of ATL formulae can be given via the following clauses:

$M, q \models p$ iff $p \in \pi(q)$ (for $p \in \Pi$);

$M, q \models \neg\varphi$ iff $M, q \not\models \varphi$;

$M, q \models \varphi \wedge \psi$ iff $M, q \models \varphi$ and $M, q \models \psi$;

$M, q \models \langle\langle A \rangle\rangle \bigcirc \varphi$ iff there is a collective strategy S_A such that, for every $\lambda \in out(q, S_A)$, we have $M, \lambda[1] \models \varphi$;

$M, q \models \langle\langle A \rangle\rangle \Box \varphi$ iff there exists S_A such that, for every $\lambda \in out(q, S_A)$, we have $M, \lambda[i]$ for every $i \geq 0$;

$M, q \models \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$ iff there exists S_A such that for every $\lambda \in out(q, S_A)$ there is an $i \geq 0$, for which $M, \lambda[i] \models \psi$, and $M, \lambda[j] \models \varphi$ for every $0 \leq j < i$.

Example 4 *Consider a simple formalization of the scenario from Example 1, presented in Figure 2.2.1A. First, the banker sets the code to either* 0 *or* 1*, and walks away. Then, the robber tries to open the safe by typing a number. If the number is correct, the safe opens; otherwise the robber is jailed in the vault. Nodes in the graph represent global states of the system. Transitions are labeled by combinations of actions from* b, r*, and* nop *stands for "no operation" or "do nothing" (formally,* nop *is just another action).*

ATL addresses agents with perfect information, so the following naturally holds: $M_1, q_0 \models \langle\langle r \rangle\rangle \Diamond$open*. The right strategy for the robber is to wait first to see which code is set, and then to type the appropriate number:* $s_r(q_0) = nop$*,* $s_r(q_1) = type0$*, and* $s_r(q_2) = type1$*.*

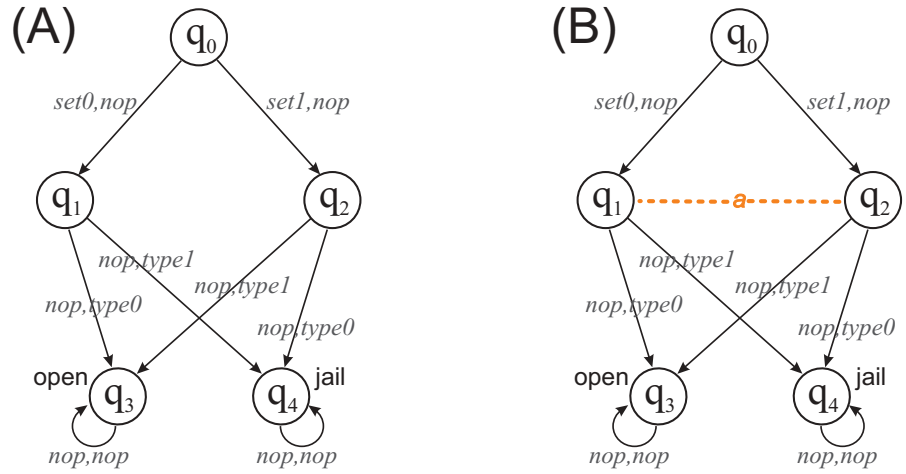


Figure 2.1: The banker and the robber: (A) concurrent game structure M_1 for the perfect information case; (B) concurrent epistemic game structure M_2 for the imperfect information case

Remark 2 *Concurrent game structures model actions as abstract atomic entities, with no underlying structure. This is not necessarily satisfying for everyone's purposes. One may, e.g., want to define actions as state transformations that can occur in the system, like in models of dynamic logic [60]; STIT models assign actions/choices with even more complicated conceptual structure [17]. We choose, after [8, 121, 83, 3], to avoid the discussion on the nature of actions, and make the simplifying assumption that actions are identified by unique names. Note that this approach follows closely the tradition of game theory, and the definition of an extensive game form in particular [111].*

One of the most appreciated features of ATL is its model checking complexity – linear in the number of transitions in the model and the length of the formula. The model checking problem is, given a formula φ and a model M with a state q, to decide whether $M, q \models \varphi$ or not.

Proposition 1 ([8]) *The ATL model checking problem is PTIME-complete, and can be done in time* $\mathbf{O}(ml)$*, where* m *is the number of transitions in the model and* l *is the length of the formula.*

Note that the complexity is measured, as usual, as a function of the size of the input. Thus, while infinite concurrent game structures make perfect sense in general, they cannot be subjects of model checking unless represented in a finite way.

Remark 3 *The result in Proposition 1 does not seem so unambiguously optimistic after a closer inspection, i.e., when we measure the size of models in the number of states, actions and agents [78, 95, 82], or when we represent systems with so called concurrent programs [131]. This remark is only meant as a note of warning; such a detailed complexity analysis for the logics of ability under imperfect information (that are the main topic here) is beyond the scope of this paper.*

2.2.2 ATL with Epistemic Logic

ATL is unrealistic in a sense: real-life agents seldom possess complete information about the current state of the world. On the other hand, imperfect information and knowledge are handled in epistemic logic in a natural way. A combination of ATL and epistemic logic, called *Alternating-time Temporal Epistemic Logic* (ATEL), was introduced by van der Hoek and Wooldridge in [133, 134] to enable reasoning about agents acting under imperfect information.

ATEL enriches the picture with an epistemic component, adding to ATL operators for representing agents' knowledge: $K_a\varphi$ reads as "agent a knows that φ". Additional operators $E_A\varphi$, $C_A\varphi$, and $D_A\varphi$, where A is a set of agents, refer to *mutual knowledge* ("everybody knows"), *common knowledge*, and *distributed knowledge* among the agents from A. Thus, $E_A\varphi$ means that every agent in A knows that φ holds, while $C_A\varphi$ means not only that the agents from A know that φ, but they also know that they know it, and that they know that they know that they know it, etc. The distributed knowledge modality $D_A\varphi$ expresses that if the agents could share their individual information they would be able to recognize that φ.

Models for ATEL extend concurrent game structures with epistemic accessibility relations $\sim_1, ..., \sim_k \subseteq Q \times Q$ (one per agent) for modeling agents' uncertainty.[4] We will call such models *concurrent epistemic game structures* (CEGS) in the rest of the paper. Agent a's epistemic relation is meant to encode a's inability to distinguish between the (global) system states: $q \sim_a q'$ means that, while the system is in state q, agent a cannot determine whether it is in q or q'. Then, the semantics of K_a is defined as:

$M, q \models K_a\varphi$ iff $M, q' \models \varphi$ for every q' such that $q \sim_a q'$.

Example 5 *Consider model M_2 from Figure 2.2.1B, with the epistemic link between states q_1 and q_2 (we omit the reflexive indistinguishability links from q_0 to q_0, q_1 to q_1 etc. to make the figure easier to read). This time, the scenario is more realistic: the robber does not know the correct code. Thus, one cannot expect him to be able to open the safe. Still, in ATEL, we have that $M_2, q_0 \models \langle\langle r \rangle\rangle \Diamond \mathsf{open}$; the same (non-uniform) strategy as in Example 4 can be used to demonstrate this. Moreover, we have even that $M_2, q_0 \models K_r \langle\langle r \rangle\rangle \Diamond \mathsf{open}$: using knowledge operators does not help, because cooperation modalities are still underpinned by a notion of strategy that does not agree with imperfect information of agents. This is a fundamental problem with ATEL, which we discuss briefly in Section 2.2.3.*

Relations $\sim_A^E$, $\sim_A^C$ and $\sim_A^D$, used to model group epistemics, are derived from the individual relations of agents from A. First, $\sim_A^E$ is the union of relations $\sim_a$, $a \in A$. Next, $\sim_A^C$ is defined as the transitive closure of $\sim_A^E$. Finally, $\sim_A^D$ is the intersection of all the $\sim_a$, $a \in A$. The semantics of group knowledge can be defined as below (for $\mathcal{K} = C, E, D$):

$M, q \models \mathcal{K}_A\varphi$ iff $M, q' \models \varphi$ for every q' such that $q \sim_A^{\mathcal{K}} q'$.

Note that $K_a \equiv C_{\{a\}} \equiv E_{\{a\}} \equiv D_{\{a\}}$, so individual knowledge operators K_a are actually redundant.

[4] The relations are assumed to be equivalences.

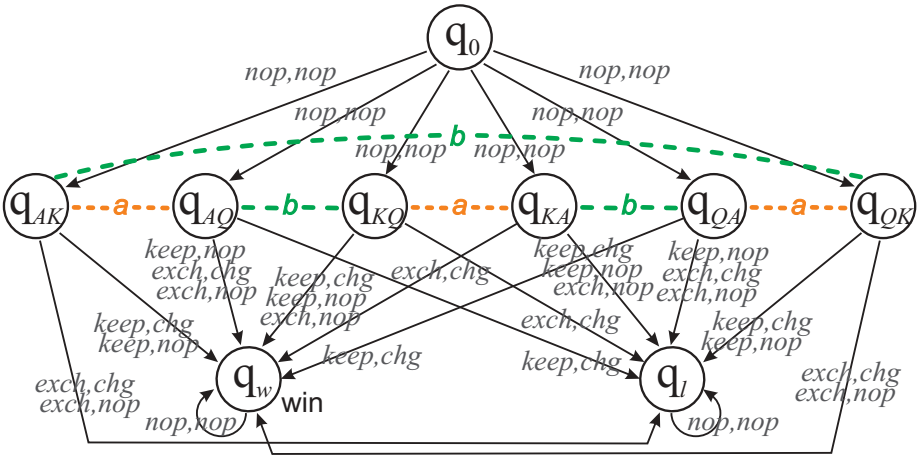


Figure 2.2: Gambling Robots game. Nodes represent global states of the system; arrows denote transitions, labeled with combinations of actions from all the agents. Dashed lines indicate states that are indistinguishable for respective agents. Actions of the environment agent are omitted from the picture to make it easier to read. As epistemic relations are by definition reflexive, we omit reflexive epistemic links too

In order to explore the subtleties of collective play, we extend the model from Figure 2.2.1B slightly: the pattern is the same, but more complex properties can be demonstrated.

Example 6 (Gambling robots) *Two robots (a and b) play a simple card game. The deck consists of Ace, King and Queen (A, K, Q). Normally, it is assumed that A is the best card, K the second best, and Q the worst; so, A beats K and Q, K beats Q, and Q beats no card. At the beginning of the game, the "environment" agent deals a random card to both robots (actions $deal_{AK}, deal_{AQ}, \ldots, deal_{QK}$), so that each player can see his own card, but he does not know the card of the other player. Then robot a can choose to exchange his card for the one remaining in the deck (action $exch$), or he can keep the current one ($keep$). At the same time, robot b can change the priorities of the cards to a Rochambeau-like game (that is, A still beats K and K beats Q, but Q becomes better than A), or he can do nothing (nop), i.e. leave the priorities unchanged. If a has a better card than b after that, then a win is scored, otherwise the game ends in a "losing" state.*

A CEGS for the game is shown in Figure 2.2.2; we will refer to the model as M_3 throughout the rest of the paper. State q_0 represents the situation before, and states $q_{AK}, \ldots, q_{QK}$ after the cards have been dealt (each $q_{c_1c_2}$ stands for the situation when a has got card c_1, and b has got card c_2). Actions of the environment are omitted from the figure for the sake of readability. Similarly to the previous example, $M_3, q_0 \models \langle\langle a, b\rangle\rangle\Diamond\mathsf{win}$ (and even $M_3, q_0 \models C_{\{a,b\}}\langle\langle a, b\rangle\rangle\Diamond\mathsf{win}$), but there is no uniform *strategy to achieve this: in order to win, a must exchange his card in state q_{QK}, so he must exchange his card in q_{QA} too (if we require uniformity), and playing $exch$ in q_{QA} leads to the losing state. So, again, we have $\langle\langle a, b\rangle\rangle\Diamond\mathsf{win}$, although intuitively $\{a, b\}$ have no feasible way of ensuring a win.*

2.2.3 Problems with ATEL

It has been pointed out in several places that the meaning of ATEL formulae can be counterintuitive [66, 83, 86]. Most importantly, one would expect that an agent's ability to achieve property φ should imply that the agent has enough control and knowledge to *identify* and *execute* a strategy that enforces φ (cf. also [121]). ATEL adds to ATL the vocabulary of epistemic logic; still, in ATEL the strategic and epistemic layers are combined as if they were independent. They should be – if we do not ask whether the agents in question are able to identify and execute their strategies. They should not if we want to interpret strategies as *executable plans*, about which the agents *know* that they guarantee achieving the goal.

First of all, executable plans should not specify different actions in indistinguishable states. Most (if not all) current approaches to strategic ability under imperfect information [83, 121, 86, 135, 63], agree with the postulate from [66] that only *uniform* strategies should be considered in the semantics of $\langle\langle A\rangle\rangle$. Formally, strategy s_a is *uniform* iff $q \sim_a q'$ implies that $s_a(q) = s_a(q')$; a collective strategy S_A is uniform iff it consists of only uniform individual strategies. In other words, agents make choices with respect to their *local* (epistemic) states rather than global states of the system. Agents are assumed to know their available actions (i.e., the choices open to them), so they must have the same choices in indistinguishable states. That is, from now on we consider only models in which $q \sim_a q'$ implies $d(a, q) = d(a, q')$.

Second, it was suggested in [83] that, when reasoning about what an agent can *enforce*, it seems more appropriate to require the agent to *know his winning strategy* rather than to know only *that such a strategy exists*. This problem is closely related to the distinction between knowledge *de re* and knowledge *de dicto*, well known in the philosophy of language [117], as well as research on the interaction between knowledge and action [105, 106, 143]. One can naturally distinguish at least four different levels of strategic ability (cf. [83]):

1. Agent a has a strategy *"de re"* to enforce φ, i.e., he has an executable winning strategy and knows the strategy (he "knows how to play");
2. Agent a has a strategy *"de dicto"* to enforce φ (i.e., he knows only that *some* executable winning strategy is available);
3. Agent a has an executable strategy to enforce φ (but not necessarily even knows about it);
4. Agent a may *happen* to behave in such a way that φ is enforced. However, the behavior can have no executable specification (i.e., there might be no uniform strategy that describes it).

Obviously, (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (4), but not the other way around. We do think that *all* of these concepts can be useful for reasoning about strategic ability under imperfect information. However, we believe that (1) is particularly important and natural. Unfortunately, ATEL enables to express only ability of type (4), as Example 5 showed. Several variations on "ATL with imperfect information" have been proposed as alternatives, yet none of them seems the ultimate definitive solution. We summarize the most important proposals in the following sections.

2.2.4 First Try: ATEL with Uniform Strategies

The first attempt to cope with these problems was presented in [66], where it was proposed that only uniform strategies should be used in the semantics of cooperation modalities. "Uniform ATEL" (U-ATEL) captures ability of type (2) and (3): $\langle\langle a\rangle\rangle\varphi$ says that a has a uniform strategy to achieve φ, and $K_a\langle\langle a\rangle\rangle\varphi$ denotes having a strategy "de dicto". However, *knowing how to play* still cannot be expressed.

Example 7 *Consider model M_2 from Figure 2.2.1B, and assume that q_1 is the current state. The robber does have a uniform strategy to open the safe in one step (play type0 at q_1 and q_2, and nop elsewhere), and indeed $M_2, q_1 \models \langle\langle r\rangle\rangle\bigcirc \mathsf{open}$. He also knows that such a strategy is available, and we have $M_2, q_1 \models K_r\langle\langle r\rangle\rangle\bigcirc \mathsf{open}$ (in every state q such that $q_1 \sim_r q$, $M, q \models \langle\langle r\rangle\rangle\bigcirc \mathsf{open}$). Still, the robber does not know how to play in q_1 to achieve* open*, and this property has no U-ATEL counterpart. Note also that $M_2, q_0 \models \neg\langle\langle r\rangle\rangle\Diamond\mathsf{open} \wedge \langle\langle\emptyset\rangle\rangle\bigcirc\langle\langle r\rangle\rangle\Diamond\mathsf{open}$, and $M_2, q_0 \models \neg K_r\langle\langle r\rangle\rangle\Diamond\mathsf{open} \wedge K_r\langle\langle\emptyset\rangle\rangle\bigcirc K_r\langle\langle r\rangle\rangle\Diamond\mathsf{open}$ (the robber has no strategy to open the safe in q_0, but he can simply wait a moment, and he will magically get one), which suggests that one should be careful when talking about abilities of type (2) and (3).*

Likewise, for the gambling robots we have $M_3, q_0 \models \neg\langle\langle a\rangle\rangle\Diamond\mathsf{win}$, and even $M_3, q_0 \models \neg\langle\langle a, b\rangle\rangle\Diamond\mathsf{win}$ (see Section 2.2.2). On the other hand, $M_3, q_{AK} \models \langle\langle a\rangle\rangle\Diamond\mathsf{win} \wedge K_a\langle\langle a\rangle\rangle\Diamond\mathsf{win}$.

2.2.5 Aggregating Initial States: "Feasible Atel"

"Feasible ATEL" [86], which we will sometimes call F-ATEL, is an update of ATEL, in which the "perfect information" cooperation modalities are kept, but the language is extended with new modalities: $\langle\langle A\rangle\rangle^f$, $\langle\langle A\rangle\rangle^f_E$, $\langle\langle A\rangle\rangle^f_C$, $\langle\langle A\rangle\rangle^f_{K_a}$ and $\langle\langle A\rangle\rangle^f_{M_a}$, that represent agents' ability to find a suitable uniform strategy, with the semantics summarized below:

$M, q \models \langle\langle A\rangle\rangle^f\bigcirc\varphi$ iff there is a uniform collective strategy S_A such that, for every $\lambda \in out(q, S_A)$, we have $M, \lambda[1] \models \varphi$.
For $\langle\langle A\rangle\rangle^f\Box\varphi$ and $\langle\langle A\rangle\rangle^f\varphi\,\mathcal{U}\,\psi$: analogously;

$M, q \models \langle\langle A\rangle\rangle^f_E\bigcirc\varphi$ iff there is a uniform collective strategy S_A such that, for every q' such that $q \sim^E_A q'$, and for every $\lambda \in out(q', S_A)$, we have $M, \lambda[1] \models \varphi$.
For $\langle\langle A\rangle\rangle^f_E\Box\varphi$ and $\langle\langle A\rangle\rangle^f_E\varphi\,\mathcal{U}\,\psi$: analogously;

$M, q \models \langle\langle A\rangle\rangle^f_C\bigcirc\varphi$ iff there is a uniform collective strategy S_A such that, for every q' such that $q \sim^C_A q'$, and for every $\lambda \in out(q', S_A)$, we have $M, \lambda[1] \models \varphi$.
For $\langle\langle A\rangle\rangle^f_C\Box\varphi$ and $\langle\langle A\rangle\rangle^f_C\varphi\,\mathcal{U}\,\psi$: analogously;

$M, q \models \langle\langle A\rangle\rangle^f_{K_a}\bigcirc\varphi$ iff there is a uniform collective strategy S_A such that, for every q' such that $q \sim_a q'$, and every $\lambda \in out(q', S_A)$, we have $M, \lambda[1] \models \varphi$.
For $\langle\langle A\rangle\rangle^f_{K_a}\Box\varphi$ and $\langle\langle A\rangle\rangle^f_{K_a}\varphi\,\mathcal{U}\,\psi$: analogously;

$M, q \models \langle\langle A\rangle\rangle^{f}_{M_a} \bigcirc \varphi$ iff there is a uniform collective strategy S_A and state q' with $q \sim_a q'$, such that, for every $\lambda \in out(q', S_A)$, we have $M, \lambda[1] \models \varphi$.
For $\langle\langle A\rangle\rangle^{f}_{M_a} \Box\varphi$ and $\langle\langle A\rangle\rangle^{f}_{M_a} \varphi \mathcal{U} \psi$: analogously.

The idea of cooperation modalities with subscripts that indicate the epistemic "mode", in which coalition A can identify their winning strategy, was further developed in the logic of ATOL, which we present in Section 2.2.6.

We note that "Uniform ATEL" can be seen as a subset of "Feasible ATEL", as the meaning of $\langle\langle A\rangle\rangle\varphi$ proposed in [66] is, for agents playing memoryless strategies, equivalent to $\langle\langle A\rangle\rangle^{f}\varphi$ from [86].

2.2.6 Going for Expressive Power: ATOL

Alternating-time Temporal Observational Logic (ATOL), proposed in [83], follows the same perspective, but it offers a richer language of strategic operators to express subtle differences between various kinds of collective abilities of teams. In this paper, we use the notation proposed in [84]. The informal meaning of $\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\varphi$ is: "group A has a (memoryless uniform) strategy to enforce φ, and agents Γ can identify the strategy as successful for A in the epistemic sense $\mathcal{K}$". For instance, $M, q \models \langle\langle A\rangle\rangle_{D(\Gamma)}\varphi$ iff there is S_A such that, for every q' with $q \sim^{D}_{\Gamma} q'$, and every $\lambda \in out(q', S_A)$, we have that φ is true for λ.

Formally, let $\mathcal{K} = E, C, D$. The semantics of the enhanced cooperation modalities can be defined as follows:

$M, q \models \langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)} \bigcirc \varphi$ iff there is a collective memoryless uniform strategy S_A such that, for every q' with $q \sim^{\mathcal{K}}_{\Gamma} q'$, and every $\lambda \in out(q', S_A)$, we have that $M, \lambda[1] \models \varphi$.
For $\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)} \Box\varphi$ and $\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)} \varphi \mathcal{U} \psi$: analogously.

Example 8 *Coming back to our gambling robots, it is easy to see that* $M_3, q_0 \models \neg\langle\langle a\rangle\rangle_{K(a)} \Diamond \mathsf{win}$*, because, for every* a*'s (uniform) strategy, if it guarantees a win in e.g. state* q_{AK} *then it fails in* q_{AQ} *(and similarly for other pairs of indistinguishable states). Let us also observe that* $M_3, q_0 \models \neg\langle\langle a, b\rangle\rangle_{E(\{a,b\})} \Diamond \mathsf{win}$*: in order to win,* a *must exchange his card in state* q_{QK}*, so he must exchange his card in* q_{QA} *too (by uniformity), and playing* $exch$ *in* q_{QA} *leads to the losing state. On the other hand,* $M_3, q_{AQ} \models \langle\langle a, b\rangle\rangle_{E(\{a,b\})} \bigcirc \mathsf{win}$ *(a winning strategy:* $s_a(q_{AK}) = s_a(q_{AQ}) = s_a(q_{KQ}) = keep$*,* $s_b(q_{AQ}) = s_b(q_{KQ}) = s_b(q_{AK}) = nop$*;* q_{AK}, q_{AQ}, q_{QK} *are the states that must be considered by* a *and* b *in* q_{AQ}*). Still,* $M_3, q_{AK} \models \neg\langle\langle a, b\rangle\rangle_{E(\{a,b\})} \bigcirc \mathsf{win}$.

ATOL allows us to express other ways of identifying a winning strategy too: we have that $M_3, q_{AK} \models \langle\langle a, b\rangle\rangle_{D(\{a,b\})} \bigcirc \mathsf{win} \wedge \langle\langle a, b\rangle\rangle_{K(a)} \bigcirc \mathsf{win}$ *(the robots can identify the strategy if they share their views of the world; also,* a *can be the "boss" who points out the strategy), and* $M_3, q_{AQ} \models \neg\langle\langle a, b\rangle\rangle_{C(\{a,b\})} \bigcirc \mathsf{win}$ *(despite both* a, b *knowing the winning strategy, they do not have common knowledge about it).*

ATOL is quite expressive. However, it does not allow for combination of strategic ability and *arbitrary* epistemic modes – the operators $\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}$ are fixed by taking $\mathcal{K} \in \{C, E, D\}$. For example, $\langle\langle A\rangle\rangle_{E_A E_A}\varphi$ is *not* a well formed ATOL formula – although it is easy to give an interpretation of such a formula in a similar manner to the other ATOL operators. Furthermore, the trebly parameterized cooperation modalities are rather baroque.

2.2.7 Elegance and Simplicity: ATL$_{ir}$

Schobbens [121] approached the problem of combining strategies with uncertainty on a more abstract level. He suggested that it makes sense to talk about agents with perfect as well as imperfect information on one hand, and perfect vs. imperfect recall on the other – and that these two fundamental semantic choices are orthogonal. This gives rise to four different logics of strategic ability: ATL$_{IR}$ (for perfect **I**nformation and perfect **R**ecall, i.e. the original ATL), ATL$_{iR}$ (for imperfect **i**nformation and perfect **R**ecall), etc. As we focus on imperfect information and memoryless strategies in this paper, the logic of ATL$_{ir}$ is most interesting for us.

Informally, $\langle\!\langle A\rangle\!\rangle_{ir}\varphi$ holds in M, q iff there is a uniform collective strategy S_A such that, for every agent $a \in A$, state q' with $q \sim_a q'$, and path $\lambda \in out(q', S_A)$, we have that φ is true for λ. In other words, there is a strategy such that *everybody in A knows* that executing this strategy will bring about φ. Formally:

$M, q \models \langle\!\langle A\rangle\!\rangle_{ir}\bigcirc\varphi$ iff there is a uniform collective strategy S_A such that, for every $a \in A$, q' such that $q \sim_A^E q'$, and path $\lambda \in out(S_A, q')$, we have $M, \lambda[1] \models \varphi$.
For $\langle\!\langle A\rangle\!\rangle_{ir}\Box\varphi$ and $\langle\!\langle A\rangle\!\rangle_{ir}\varphi\,\mathcal{U}\,\psi$: analogously.

Example 9 *For our gambling robots, we get e.g. that:* $M_3, q_0 \models \neg\langle\!\langle a\rangle\!\rangle_{ir}\Diamond\mathsf{win}$, $M_3, q_0 \models \neg\langle\!\langle a, b\rangle\!\rangle_{ir}\Diamond\mathsf{win}$, $M_3, q_{AQ} \models \langle\!\langle a, b\rangle\!\rangle_{ir}\bigcirc\mathsf{win}$, *and* $M_3, q_{AK} \models \neg\langle\!\langle a, b\rangle\!\rangle_{ir}\bigcirc\mathsf{win}$.

Note that $\langle\!\langle A\rangle\!\rangle_{ir}\Phi$ is equivalent to the "Feasible ATEL" formula $\langle\!\langle A\rangle\!\rangle_E^f\Phi$, and the ATOL formula $\langle\!\langle A\rangle\!\rangle_{E(A)}\Phi$. Moreover, it is not possible to express in ATL$_{ir}$ that A have common knowledge about the successful strategy, or that they are able to identify it if they share their information etc. On the other hand, ATL$_{ir}$ stands out among the existing proposals for its simplicity and conceptual clarity, and can be treated as the "core", minimal ATL-based language for ability under imperfect information.

The following proposition sums up some of the results presented in [121, 81, 83]:

Proposition 2 *Model checking "Feasible ATEL", ATL$_{ir}$ and ATOL is $\mathbf{\Delta_2^P}$-complete in the number of transitions (and epistemic links) in the model, and the length of the formula.*

In Section 2.3, we will propose Constructive Strategic Logic (CSL) which strictly subsumes ATOL, while sharing (in our opinion) the elegance of ATL$_{ir}$, and model checking complexity of all of the approaches discussed above. The main idea behind CSL is that we would like to express various levels of ability with combinations of *some* kind of epistemic operators with *some* kind of cooperation modalities. Before we present our proposal, we want to mention two logics that, to a limited extent, have achieved a similar trait. The logics are briefly presented in Sections 2.2.8 and 2.2.9.

2.2.8 Abilities of Rational Players: ETSL

Epistemic Temporal Strategic Logic [135] digs deeper in the repository of game theory, and focuses on the concept of *undominated strategies*. Its variant of the cooperation modalities has a different flavor than the ones from ATL, ATEL, ATOL etc. In a way, $\langle\langle A\rangle\rangle\varphi$ in ETSL can be summarized as: "if A play *rationally* to achieve φ (meaning: they never play a dominated strategy), they will achieve φ".

ETSL is underpinned by several interesting concepts. Unfortunately, its original semantics from [135] comes with a plethora of auxiliary functions and definitions (and a couple of omissions), which make it rather hard to read. Moreover, the semantics is defined only for *finite turn-based acyclic* game models, and the satisfaction relation refers not only to models and states (respectively paths), but also to a fixed strategy $S_{\mathbb{A}\mathrm{gt}}$ (assumed to represent the current strategies of all agents). It has been shown in [68], that the semantics can be extended to concurrent epistemic game structures, and given in a more compact way. Moreover, for "vanilla" ETSL formulae,[5] it can be given via standard semantic clauses for state formulae.

Let M be a CEGS. First, we define the notion of domination as follows. Let $\Phi \equiv \bigcirc\psi$, $\Box\psi$, or $\psi_1\,\mathcal{U}\,\psi_2$, where ψ, ψ_1, ψ_2 are "vanilla" ETSL formulae. Moreover, let $|\Phi|$ denote the set of paths for which Φ holds; formally, $|\bigcirc\psi| = \{\lambda \mid M, \lambda[1] \models \psi\}$, $|\Box\psi| = \{\lambda \mid \forall_i M, \lambda[i] \models \psi\}$, and $|\psi_1\,\mathcal{U}\,\psi_2| = \{\lambda \mid \exists_i (M, \lambda[i] \models \psi_2 \wedge \forall_{0\leq j<i} M, \lambda[j] \models \psi_1\}$. Then, *strategy S_A dominates strategy T_A wrt. $M, q,$ and Φ* iff both of the following conditions hold:

1. for every q' with $q \sim^E_A q'$: if $out(q', T_A) \subseteq |\Phi|$ then also $out(q', S_A) \subseteq |\Phi|$;
2. there is q' such that $q \sim^E_A q'$, and $out(q', S_A) \subseteq |\Phi|$, and $out(q', T_A) \not\subseteq |\Phi|$.

Strategy S_A is *undominated* wrt. M, q, Φ iff there is no strategy that dominates S_A wrt M, q, Φ.

Now the semantics of $\langle\langle A\rangle\rangle$ in ETSL can be expressed entirely in terms of models and their states:

$M, q \models \langle\langle A\rangle\rangle\bigcirc\varphi$ iff for every strategy S_A, undominated wrt $M, q, \bigcirc\varphi$, and every $\lambda \in out(q, S_A)$, we have that $M, \lambda[1] \models \varphi$.
For $\langle\langle A\rangle\rangle\Box\varphi$ and $\langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\psi$: analogously.

The relationship between ETSL and Constructive Strategic Logic is briefly discussed in Section 2.4.4. We conjecture that neither of them subsumes the other, but there are several interesting associations. The most interesting feature of ETSL is perhaps the fact that, by combining standard epistemic operators and its non-standard cooperation modalities, we can capture "knowing how to play" for individual agents (although this does not extend to collective agents), see [68] or Section 2.4.4 for more details.

2.2.9 Explicit Actions: ATEL-A

In AT(E)L, it is not possible to refer directly to particular actions in the logical language. For example, it is not possible to express the fact that "if agent

[5] *I.e,* formulae in which every temporal operator is preceded by exactly one cooperation modality.

i chooses action α, then formula φ will necessarily be true in the next moment". ATEL-A [3] allows such expressions by introducing names of actions, in addition to names of agents, inside cooperation modalities. For instance, the above expression can be written as $\langle\langle \alpha_i \rangle\rangle \bigcirc \varphi$. This makes it possible to capture the levels of ability, discussed in Section 2.2.3, in the limited case of properties that can be achieved in one step:

(4), (3) $\langle\langle i \rangle\rangle \bigcirc \varphi$: agent i may behave in such a way that φ is enforced next. Note that there is no difference between (4) and (3) when we only talk about the next state – then uniformity does not play any role;

(2) $K_i \langle\langle i \rangle\rangle \bigcirc \varphi$: agent i has a strategy "de dicto" to enforce φ next;

(1) $\bigvee_{\alpha \in Act} K_i \langle\langle \alpha_i \rangle\rangle \bigcirc \varphi$: agent i has a strategy "de re" to enforce φ next.

Because of explicit actions, ATEL-A is not directly comparable to the logics considered in this paper, and we will not discuss ATEL-A further.

2.2.10 Other Possibilities

In the original formulation of ATL, agents were assumed to have perfect recall of the game, in the sense that they could base their decisions on *sequences* of states rather than on single states. Variants of ATL for perfect recall and imperfect information have also been considered, cf. ATL$_{iR}$ [121] and ATEL-R* [83]. However, as agents seldom have unlimited memory, and logics of strategic ability with imperfect information and perfect recall are believed to have undecidable model checking [8, 121], we do not investigate this variant of ability here.

Yet another, very recent, proposal [63] approaches the problem of strategic abilities under imperfect information within the framework of STIT (the logic of *seeing to it that*). STIT shares many similarities with ATL, but it comes from a different tradition, and its technical formulation is markedly different from that of ATL. Thus, in order to analyze STIT-based proposals in our new framework, one must first establish the precise relationship between both frameworks, i.e., compare models, semantics, expressive power, pragmatics (e.g., verification issues) etc. Several important results in this respect have already been reported [142, 22], but there is still much to be done.

2.3 Constructive Strategic Logic: A New Semantics for Ability and Knowledge

ATOL covers more cases than ATL$_{ir}$ and "Feasible ATEL", and it is not committed to any notion of rationality (unlike ETSL). One major drawback of ATOL is that it vastly increases the number of modal operators necessary to express properties of agents. For team A, a whole family of cooperation modalities $\langle\langle A \rangle\rangle_{\mathcal{K}(\Gamma)}$ is used (instead of a single modality $\langle\langle A \rangle\rangle$ in ATL) to specify who should identify the right strategy for A, in what way etc. It would be much more elegant to modify the semantics of "simple" cooperation modalities $\langle\langle A \rangle\rangle$ and/or epistemic operators, so that they can be composed into sufficiently expressive formulae. The problem with strategic ability under

uncertainty is that, when analyzing consequences of their strategies, agents must consider also the outcome paths starting from states other than the current state – namely, from all states that *look the same* as the current state. Thus, a property of a strategy being successful with respect to goal φ is not local to the current state; *the same* strategy must be successful in all "opening" states being considered. In order to capture this feature of strategic ability under imperfect information, we change the type of the satisfaction relation $\models$, and define what it means for a formula φ to be satisfied in a set of states $Q \subseteq St$ of model M. To our best knowledge, nobody has used this kind of semantics yet.

Moreover, we extend the language of ATEL with unary "constructive knowledge" operators $\mathbb{K}_a$, one for each agent a, that yield the set of states, indistinguishable from the current state from a's perspective. Constructive common, mutual, and distributed knowledge are formalized via operators $\mathbb{C}_A, \mathbb{E}_A$, and $\mathbb{D}_A$.

2.3.1 Language and Semantics

The language of Constructive Strategic Logic (CSL) includes atomic propositions, Boolean connectives, strategic formulae, standard epistemic operators, and *constructive knowledge operators* for groups of agents (individual knowledge can be defined as a special case of collective knowledge – see below):

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle \bigcirc \varphi \mid \langle\langle A\rangle\rangle \Box \varphi \mid \langle\langle A\rangle\rangle \varphi \mathcal{U} \varphi \mid C_A\varphi \mid E_A\varphi \mid D_A\varphi \mid \mathbb{C}_A\varphi \mid \mathbb{E}_A\varphi \mid \mathbb{D}_A\varphi.$$

where A is a set of agents.

Remark 4 *As we will show in Section 2.7.2, standard knowledge can be defined as a special kind of constructive knowledge, and therefore the standard knowledge operators do* not *have to be included in the language. However, rather than immediately deriving C_A, E_A, D_A from $\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$, we choose to give the semantic clauses for all of them, and only later prove the relationship formally.*

Models are concurrent epistemic game structures again; that is, we interpret the formulae of CSL over exactly the same class of models which was used for ATEL, ATL$_{ir}$, ATOL etc. To recapitulate, a CEGS can be defined as a tuple

$$M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o, \sim_1, ..., \sim_k \rangle,$$

where:

- $\mathbb{A}\mathrm{gt} = \{1, ..., k\}$ is a finite nonempty set of all agents,
- St is a nonempty set of states,
- Π is a set of atomic propositions,
- $\pi : St \to 2^{\Pi}$ is a valuation of propositions,
- Act is a nonempty set of (atomic) actions;

- function $d : \mathbb{A}\mathrm{gt} \times St \to 2^{Act}$ defines actions available to an agent in a state; $d(a, q) \neq \emptyset$ for all $a \in \mathbb{A}\mathrm{gt}, q \in St$,
- o is a (deterministic) transition function that assigns an outcome state to each combination of a state and a vector of actions (one action per agent). That is, $o(q, \alpha_1, \ldots, \alpha_k) \in St$ for every $q \in St$ and $\langle \alpha_1, \ldots, \alpha_k \rangle \in d(1, q) \times \cdots \times d(k, q)$;
- $\sim_1, ..., \sim_k \subseteq St \times St$ are epistemic accessibility relations, one per agent. It is assumed that each $\sim_a$ is an equivalence relation, and that $q \sim_a q'$ implies $d(a, q) = d(a, q')$.

Again, a (memoryless) strategy s_a of agent a is a conditional plan represented by function $s_a : St \to Act$ such that $s_a(q) \in d(a, q)$ for every q. A collective strategy S_A is a tuple of strategies, one per agent from A. Strategy s_a is uniform iff $q \sim_a q'$ implies $s_a(q) = s_a(q')$; a collective strategy is uniform iff it consists of only uniform individual strategies. A path λ is an infinite sequence of states that can be effected by subsequent transitions; by $\lambda[i]$, we denote the ith position on path λ. Function $out(q, S_A)$ returns the set of all paths that may result from agents A executing strategy S_A from state q onward (see Section 2.2.1 for the precise definition). Collective epistemic relations are defined as: $\sim_A^D = \bigcap_{a \in A} \sim_a$, $\sim_A^E = \bigcup_{a \in A} \sim_a$; $\sim_A^C$ is defined as the transitive closure of $\sim_A^E$.

Now we define the notion of a formula φ being satisfied by a (non-empty) set of states Q in a model M, written $M, Q \models \varphi$. We will also write $M, q \models \varphi$ as a shorthand for $M, \{q\} \models \varphi$. Note that it is the latter notion of satisfaction (in single states) that we will ultimately be interested in – but that notion is defined in terms of the (more general) satisfaction in sets of states. Let $\mathrm{img}(q, \mathcal{R})$ be the image of state q with respect to binary relation $\mathcal{R}$, i.e., the set of all states q' such that $q\mathcal{R}q'$. Moreover, we use $out(Q, S_A)$ as a shorthand for $\bigcup_{q \in Q} out(q, S_A)$, and $\mathrm{img}(Q, \mathcal{R})$ as a shorthand for $\bigcup_{q \in Q} \mathrm{img}(q, \mathcal{R})$. The new semantics is given through the following clauses. In the semantics of cooperation modalities, only memoryless uniform strategies are considered.

$M, Q \models p$ iff $p \in \pi(q)$ for every $q \in Q$;

$M, Q \models \neg\varphi$ iff $M, Q \not\models \varphi$;

$M, Q \models \varphi \wedge \psi$ iff $M, Q \models \varphi$ and $M, Q \models \psi$;

$M, Q \models \langle\langle A \rangle\rangle \bigcirc \varphi$ iff there exists S_A such that, for every $\lambda \in out(Q, S_A)$, we have that $M, \{\lambda[1]\} \models \varphi$;

$M, Q \models \langle\langle A \rangle\rangle \Box \varphi$ iff there exists S_A such that, for every $\lambda \in out(Q, S_A)$ and $i \geq 0$, we have $M, \{\lambda[i]\} \models \varphi$;

$M, Q \models \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$ iff there exists S_A such that, for every $\lambda \in out(Q, S_A)$, there is an $i \geq 0$ for which $M, \{\lambda[i]\} \models \psi$ and $M, \{\lambda[j]\} \models \varphi$ for every $0 \leq j < i$.

$M, Q \models \mathcal{K}_A \varphi$ iff $M, q \models \varphi$ for every $q \in \mathrm{img}(Q, \sim_A^{\mathcal{K}})$ (where $\mathcal{K} = C, E, D$).

$M, Q \models \hat{\mathcal{K}}_A \varphi$ iff $M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}) \models \varphi$ (where $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$ and $\mathcal{K} = C, E, D$, respectively).

The satisfaction relation $\models$ gives us both the traditional notion of satisfaction in a state, and the more general notion of satisfaction in a set of states. As mentioned above, we are usually interested in the former, but in order to interpret, e.g., an expression such as $\mathbb{C}_A\langle\langle A\rangle\rangle\bigcirc p$ in a single state, we must interpret the subexpression $\langle\langle A\rangle\rangle\bigcirc p$ in a set of states.

Formally, the language includes only operators for representing knowledge of teams. However, individual knowledge operators can be defined in the usual manner as:

$$\begin{aligned} K_a\varphi &\equiv C_{\{a\}}\varphi, \text{ and} \\ \mathbb{K}_a\varphi &\equiv \mathbb{C}_{\{a\}}\varphi. \end{aligned}$$

As a brief example, take the formula $\varphi = \mathbb{K}_a\langle\langle a\rangle\rangle\bigcirc\psi$ where a is an agent. We have that $M, q \models \varphi$ iff there is a strategy S_a for a such that for every $\lambda \in out(\mathrm{img}(q, \sim_a), S_a)$, $M, \lambda[1] \models \psi$; in other words iff there is an (executable) strategy for a which is successfull (achieves ψ in the next state) in all the states that a considers to be possible. Or, in the terminology of Section 2.2.3, a knows a winning strategy – a has a strategy *de re* (for achieving ψ). We will discuss how the logic captures many subtly different properties of ability under imperfect information in more detail in Section 2.4, after we have clarified a few additional fundamental issues.

We employ the usual definition of the "sometime" operator:

$$\Diamond\varphi \equiv \top\,\mathcal{U}\,\varphi$$

We will also use derived propositional connectives. However, the exact meaning of these in the non-standard semantics must be carefully studied, and we will do that in Section 2.3.2. The CSL concept of validity is discussed in Section 2.3.3.

A note on notation: as above, we will henceforth use $\mathcal{K}_A$ to denote an arbitrary standard knowledge operator for agents A (i.e., C_A, E_A or D_A), and we use $\hat{\mathcal{K}}_A$ to denote the constructive knowledge operator corresponding to $\mathcal{K}_A$, i.e., $\hat{C}_A = \mathbb{C}_A$, $\hat{E}_A = \mathbb{E}_A$ and $\hat{D}_A = \mathbb{D}_A$. We use $\mathcal{K}, \mathcal{K}', \mathcal{K}_1, \mathcal{K}_2$ etc. to denote arbitrary standard knowledge operators for arbitrary sets of agents, and, again, $\hat{\mathcal{K}}, \hat{\mathcal{K}}', \hat{\mathcal{K}}_1, \hat{\mathcal{K}}_2$ etc. to denote the corresponding constructive modalities.

2.3.2 Additional Operators

In addition to the derived operators introduced in Section 2.3.1, we use a slightly unusual definition of the Boolean "false" and "true" constants:

$$\begin{aligned} \bot &\equiv \langle\langle\emptyset\rangle\rangle(p \wedge \neg p)\,\mathcal{U}\,(p \wedge \neg p), \quad \text{where } p \text{ is an arbitrary primitive proposition,} \\ \top &\equiv \langle\langle\emptyset\rangle\rangle(\neg\bot)\,\mathcal{U}\,(\neg\bot) \end{aligned}$$

and the usual definition of Boolean connectives[6]:

$$\begin{aligned} \varphi_1 \vee \varphi_2 &\equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2), \\ \varphi_1 \rightarrow \varphi_2 &\equiv \neg\varphi_1 \vee \varphi_2, \text{ and} \\ \varphi_1 \leftrightarrow \varphi_2 &\equiv (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_2). \end{aligned}$$

The above Boolean operators have the following semantic characterizations:

Proposition 3

1. $M, Q \not\models \bot$ *for all* $Q \subseteq St$, $Q \neq \emptyset$.
2. $M, Q \models \top$ *for all* $Q \subseteq St$, $Q \neq \emptyset$.
3. $M, Q \models \varphi_1 \vee \varphi_2$ *iff* $M, Q \models \varphi_1$ *or* $M, Q \models \varphi_2$.
4. $M, Q \models \varphi_1 \rightarrow \varphi_2$ *iff* $M, Q \models \varphi_1$ *implies* $M, Q \models \varphi_2$.
5. $M, Q \models \varphi_1 \leftrightarrow \varphi_2$ *iff we have that* $M, Q \models \varphi_1$ *iff* $M, Q \models \varphi_2$.

Proof

1) Suppose that $M, Q \models \bot$ for some $Q \neq \emptyset$. Then $M, Q \models \langle\langle\emptyset\rangle\rangle(p \wedge \neg p)\,\mathcal{U}\,(p \wedge \neg p)$, so for all paths λ starting from the states in Q we have $M, \lambda[0] \models p \wedge \neg p$. That is, for all $q \in Q$: $M, q \models p \wedge \neg p$. As Q is nonempty, there is at least one such q. But that means that $p \in \pi(q)$ and $p \notin \pi(q)$, which cannot be the case.

2) Analogous.

3) $M, Q \models \varphi_1 \vee \varphi_2$ iff $M, Q \models \neg(\neg\varphi_1 \wedge \neg\varphi_2)$ iff $M, Q \not\models \neg\varphi_1 \wedge \neg\varphi_2$ iff $M, Q \not\models \neg\varphi_1$ or $M, Q \not\models \neg\varphi_2$ iff $M, Q \models \varphi_1$ or $M, Q \models \varphi_2$.

4), 5) Straightforward from the above.

■

To conclude the analysis of standard connectives in this (rather nonstandard) setting, we observe that the $\neg$ operator behaves like classical negation: it obeys the law of double negation, the law of excluded middle, and the consistency requirement in every possible context:

Proposition 4 *We have the following for every* M *and* $Q \subseteq St$*:*

1. $M, Q \models \neg\neg\varphi \leftrightarrow \varphi$,
2. $M, Q \models \varphi \vee \neg\varphi$,
3. $M, Q \models \neg(\varphi \wedge \neg\varphi)$.

[6] The reason why we use the above definitions of $\top$ and $\bot$ instead of the more common ones: $\bot \equiv p \wedge \neg p$, $\top \equiv \neg\bot$ is that in the restricted language CSL$^-$, discussed in Section 2.6.3, certain formulae are disallowed, namely the ones in which negation (or a sequence of conjunctions, followed by negation) follows a constructive knowledge operator. Defining the Boolean constants the way we do, we make sure that no unraveling of $\top$ or $\bot$ will ever lead to such a formula.

Proof Straightforward from Proposition 3 and the semantic definition of $\neg$. ■

It should be noted that there are other possibilities for defining negation, disjunction and implication, corresponding to the different ways of quantifying over the set Q. We discuss the issue in more detail in Section 2.7.

2.3.3 Validity

We say that a formula is *weakly valid* (or simply *valid*) if it is satisfied individually by *each state* in every model, i.e., if $M, q \models \varphi$ for all models M and states q in M. It is *strongly valid* if it is satisfied by all non-empty *sets* in all models; i.e., if for each M and every non-empty set of states Q it is the case that $M, Q \models \varphi$. We are ultimately interested in the former (see Remark 5 below). The importance of strong validity, on the other hand, lies in the fact that strong validity of $\varphi \leftrightarrow \psi$ makes φ and ψ completely interchangeable (cf. Proposition 6.2). It is not difficult to see that the same is not true for weak validity.

Proposition 5 *1. Strong validity implies validity.*

2. Validity does not imply strong validity.

Proof (1) Straightforward. (2) We here take the liberty to refer forward to some simple results we haven't proven yet, because it is instructive to point out the distinction between weak and strong validity at this point. By Propositions 3.5 and 18, we have that for any M and set of states Q, $M, Q \models \langle\langle\emptyset\rangle\rangle\varphi\,\mathcal{U}\,\varphi \leftrightarrow \varphi$ iff ($\forall_{q\in Q} M, q \models \varphi$ iff $M, Q \models \varphi$). It follows immediately that $\langle\langle\emptyset\rangle\rangle\varphi\,\mathcal{U}\,\varphi \leftrightarrow \varphi$ is (weakly) valid, for any φ. It follows from Lemma 1.1 that there is a M and a set of states Q and a formula φ such that $M, Q \not\models \varphi$ but $\forall_{q\in Q} M, q \models \varphi$; thus $\langle\langle\emptyset\rangle\rangle\varphi\,\mathcal{U}\,\varphi \leftrightarrow \varphi$ is not strongly valid. ■

Remark 5 *The term* the logic *is sometimes understood as the set of all valid formulae in the logic. In this sense, we define* the logic of CSL *as the set of all weakly valid formulae of CSL. In a similar way, we say that a formula φ is* CSL-satisfiable *if it is weakly satisfiable in CSL, i.e., there is a model M and a state q such that $M, q \models \varphi$.*

Propositions 3.4 and 3.5 from Section 2.3.2 have two important consequences. First, the rule of Modus Ponens is correct with respect to this semantics. Second, if $\varphi_1 \leftrightarrow \varphi_2$ is strongly valid, then formulae φ_1 and φ_1 are completely interchangeable under strong (and hence also weak) validity.

Proposition 6

1. *If $\varphi_1 \rightarrow \varphi_2$ is strongly (resp. weakly) valid, and φ_1 is strongly (resp. weakly) valid, then φ_2 is strongly (resp. weakly) valid.*
2. *If $\varphi_1 \leftrightarrow \varphi_2$ is strongly valid, and ψ' is obtained from ψ through replacing an occurrence of φ_1 by φ_2, then $M, Q \models \psi$ iff $M, Q \models \psi'$.*

Proof Straightforward. ■

2.4 Expressing Agents' Strategic Abilities

In the language of Constructive Strategic Logic, strategic properties of coalitions can be expressed in a flexible and elegant way. To support this claim, we first show that the philosophical discourse on various levels of knowledge and ability, mentioned in Section 2.2.3, has its formal counterpart in CSL formulae. Then, we present a translation of ATL$_{ir}$, ATOL and "Feasible ATEL" to CSL, and thus prove that the latter embeds the former ones. We also discuss the relationship between ETSL and CSL. To avoid confusion, we will use the satisfaction sign with subscripts ($\models_{\text{ATOL}}$, $\models_{\text{CSL}}$, $\models_{\text{ETSL}}$ etc.), indicating which semantics is currently referred to.

2.4.1 Capturing Levels of Strategic Power

The reason why we need to interpret formulae over sets of states is that we need non-standard epistemic operators: $M, q \models \mathbb{K}_a\langle\langle a\rangle\rangle\varphi$ expresses the fact that a has a single strategy that enforces φ from *all* states indiscernible from q, instead of stating that φ can be achieved from *every* such state *separately*. Note that the latter property is very much in the spirit of standard epistemic logic, and indeed can be captured with the standard knowledge operator (via $K_a\langle\langle a\rangle\rangle\varphi$). Speaking in more abstract terms:

1. $\mathbb{K}_a\langle\langle a\rangle\rangle\varphi$ refers to agent a having a strategy *"de re"* to enforce φ (i.e. having a successful strategy and knowing the strategy);

2. $K_a\langle\langle a\rangle\rangle\varphi$ refers to agent a having a strategy *"de dicto"* to enforce φ (i.e. knowing only that *some* successful strategy is available);

3. $\langle\langle a\rangle\rangle\varphi$ expresses that agent a has a strategy to enforce φ *from the current state* (but not necessarily even knows about it).

Above, each of the three formulae are informally interpreted in an assumed (single) state q of a model M, i.e., we discuss the meaning of, e.g., $M, q \models \mathbb{K}_a\langle\langle a\rangle\rangle\varphi$. The meaning of this formula in this single state is again defined by interpreting a subformula in a certain set of states. By *strategies* here, we only mean executable (i.e., uniform) strategies. Capturing different ability levels of coalitions is analogous, with various "epistemic modes" of collective recognizing the right strategy.

Example 10 *Robot a has no winning strategy in the starting state of the game: $M_3, q_0 \models \neg\langle\langle a\rangle\rangle\Diamond\mathsf{win}$, which implies that it has neither a strategy "de re" nor "de dicto": $M_3, q_0 \models \neg\mathbb{K}_a\langle\langle a\rangle\rangle\Diamond\mathsf{win} \wedge \neg K_a\langle\langle a\rangle\rangle\Diamond\mathsf{win}$. On the other hand, he has a successful strategy in q_{AK} (just play $keep$) and it knows it has one (because another action, $exch$, is bound to win in q_{AQ}); still, the knowledge is not constructive, since a does not know which strategy is the right one in the current situation: $M_3, q_{AK} \models \langle\langle a\rangle\rangle\bigcirc\mathsf{win} \wedge K_a\langle\langle a\rangle\rangle\bigcirc\mathsf{win} \wedge \neg\mathbb{K}_a\langle\langle a\rangle\rangle\bigcirc\mathsf{win}$.*

Other properties of the gambling robots, that we discussed in Examples 9 and 8, can be easily expressed in the new logic by combining constructive knowledge with cooperation modalities: $M_3, q_0 \models \neg\mathbb{E}_{\{a,b\}}\langle\langle a, b\rangle\rangle\Diamond\mathsf{win}$, $M_3, q_{AK} \models \mathbb{D}_{\{a,b\}}\langle\langle a, b\rangle\rangle\bigcirc\mathsf{win} \wedge \mathbb{K}_a\langle\langle a, b\rangle\rangle\bigcirc\mathsf{win} \wedge \neg\mathbb{E}_{\{a,b\}}\langle\langle a, b\rangle\rangle\bigcirc\mathsf{win}$, $M_3, q_{AQ} \models \mathbb{E}_{\{a,b\}}\langle\langle a, b\rangle\rangle\bigcirc\mathsf{win} \wedge \neg\mathbb{C}_{\{a,b\}}\langle\langle a, b\rangle\rangle\bigcirc\mathsf{win}$ etc. In fact, it turns out that the new logic is expressive enough

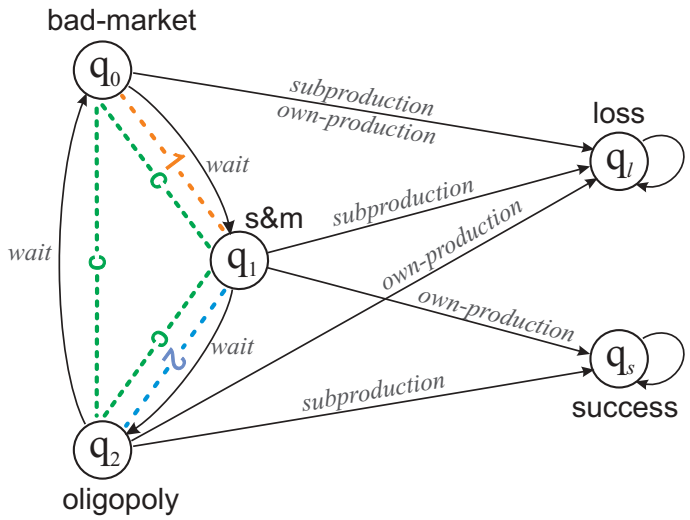


Figure 2.3: Simple market example: model M_4

to embed most approaches we have discussed. We present an appropriate translation in the next section.

Example 11 *Consider a market model, depicted in Figure 2.4.1, which formalizes in a very simple way the scenario from Example 3. The economy is assumed to run in simple cycles: after the moment of bad economy (*$\mathsf{bad-market}$*), there is always a good time for small and medium enterprises (*$\mathsf{s\&m}$*), after which the market tightens and an oligopoly emerges. At the end, the market gets stale, and we have stagnation and bad economy again.*

The company c is the only agent whose actions are represented in the model. The company can wait (action wait) or decide to start production: either on its own (own-production), or as a subcontractor of a major company (subproduction). Both decisions can lead to either loss or success, depending on the current market conditions. However, the company management cannot recognize the market conditions: bad market, time for small and medium enterprises, and oligopoly market look the same to them, as the epistemic links for c indicate.

The company can call the services of two marketing experts. Expert 1 is a specialist on oligopoly, and can recognize oligopoly conditions (although she cannot distinguish between bad economy and s&m market). Expert 2 can recognize bad economy, but he cannot distinguish between other types of market. The experts' actions have no influence on the actual transitions of the model, and are omitted from the graph in Figure 2.4.1. It is easy to see that the company cannot identify a successful strategy on its own: for instance, for the small and medium enterprises period, we have that $M_4, q_1 \models \neg\mathbb{K}_c\langle\langle c\rangle\rangle\Diamond\mathsf{success}$. It is not even enough to call the help of a single expert: $M_4, q_1 \models \neg\mathbb{K}_1\langle\langle c\rangle\rangle\Diamond\mathsf{success} \wedge \neg\mathbb{K}_2\langle\langle c\rangle\rangle\Diamond\mathsf{success}$, or to ask the experts to independently work out a common strategy: $M_4, q_1 \models \neg\mathbb{E}_{\{1,2\}}\langle\langle c\rangle\rangle\Diamond\mathsf{success}$. Still, the experts can propose the right strategy if they join forces and cooperate to find the solution: $M_4, q_1 \models \mathbb{D}_{\{1,2\}}\langle\langle c\rangle\rangle\Diamond\mathsf{success}$.

Note that this is not true any more for bad market. That is, $M_4, q_0 \models \neg\mathbb{D}_{\{1,2\}}\langle\langle c\rangle\rangle\Diamond\mathsf{success}$, because c is a memoryless agent, and it has no

uniform strategy to enforce success *from* q_0 *at all. However, the experts can suggest a more complex scheme that involves consulting them once again in the future:* $M_4, q_0 \models \mathbb{D}_{\{1,2\}}\langle\langle c\rangle\rangle\bigcirc\mathbb{D}_{\{1,2\}}\langle\langle c\rangle\rangle\Diamond$success.

For strategic abilities, standard knowledge corresponds to knowing "de dicto", while constructive knowledge captures "knowing how to play". We observe that both kinds of epistemic operators can be combined in a meaningful way. For example, $K_a\mathbb{K}_b\langle\langle b\rangle\rangle\Diamond$win says that agent a knows that player b knows how to win. Note that this is substantially different from $\mathbb{K}_a\mathbb{K}_b\langle\langle b\rangle\rangle\Diamond$win, which says that agent a can identify a strategy which b knows to be winning. Also, when interleaving epistemic operators with strategic operators, we can, e.g., describe an ability to acquire, distribute or maintain *ability*. For instance, $\mathbb{K}_a\langle\langle a\rangle\rangle\Box\mathbb{K}_b\langle\langle b\rangle\rangle\Diamond$win means that a knows how to maintain b's (constructive) ability to win, while $K_a\langle\langle a\rangle\rangle\Box\mathbb{K}_b\langle\langle b\rangle\rangle\Diamond$win says only that a knows that this is in principle possible, and $\mathbb{K}_a\langle\langle a\rangle\rangle\Box K_b\langle\langle b\rangle\rangle\Diamond$win says that a knows how to keep b aware that a winning strategy exists.

2.4.2 Expressivity of CSL

Let $\mathcal{L}$ be the logic of ATL$_{ir}$, ATOL or F-ATEL, and let φ, ψ be formulae of $\mathcal{L}$. Also, let $\mathcal{K} = C, E, D$ and $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$, respectively. Then, let the translation function tr be defined as follows:

$$
\begin{array}{cc}
tr(p) = p & tr(\neg\varphi) = \neg tr(\varphi) \\
tr(\varphi \wedge \psi) = tr(\varphi) \wedge tr(\psi) & tr(\bigcirc\varphi) = \bigcirc tr(\varphi) \\
tr(\Box\varphi) = \Box tr(\varphi) & tr(\varphi\,\mathcal{U}\,\psi) = tr(\varphi)\,\mathcal{U}\,tr(\psi) \\
tr(\langle\langle A\rangle\rangle_{ir}\varphi) = \mathbb{E}_A\langle\langle A\rangle\rangle tr(\varphi) & tr(\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\varphi) = \hat{\mathcal{K}}_\Gamma\langle\langle A\rangle\rangle tr(\varphi) \\
tr(\langle\langle A\rangle\rangle^{f}\varphi) = \langle\langle A\rangle\rangle tr(\varphi) & tr(\langle\langle A\rangle\rangle^{f}_{\mathcal{K}}\varphi) = \hat{\mathcal{K}}_A\langle\langle A\rangle\rangle tr(\varphi) \\
tr(\langle\langle A\rangle\rangle^{f}_{K_b}\varphi) = \mathbb{K}_b\langle\langle A\rangle\rangle tr(\varphi) & tr(\langle\langle A\rangle\rangle^{f}_{M_b}\varphi) = \neg K_b\neg\langle\langle A\rangle\rangle tr(\varphi) \\
tr(\mathcal{K}_A\varphi) = \mathcal{K}_A tr(\varphi) &
\end{array}
$$

The following result justifies the translation.

Theorem 1 $M, q \models_{\mathcal{L}} \varphi$ *iff* $M, q \models_{CSL} tr(\varphi)$.

Proof in the Appendix.

Corollary 1 *The translation yields a reduction of ATL$_{ir}$, ATOL and "Feasible ATEL" model checking problems to CSL model checking. The time needed for the reduction, and the resulting formula, are linear in the length of the original formula. We summarize the model checking complexity results for CSL in Section 2.5.*

Proposition 7 *Constructive Strategic Logic is* strictly more expressive *than ATL$_{ir}$, ATOL etc.*

Proof It is sufficient to prove that there is a CSL formula φ that has no ATOL equivalent (i.e., there is no ATOL formula which holds in exactly the same models and states as φ). Consider the formula $\varphi \equiv \mathbb{E}_A\mathbb{E}_A\langle\langle A\rangle\rangle\psi$. For most models ($M_3$ from Figure 2.2.2 being an example) we have $\sim^D_A \subsetneq \sim^E_A \subsetneq \sim^E_A \circ \sim^E_A \subsetneq \sim^C_A$, so φ is equivalent to neither $\langle\langle A\rangle\rangle_{D(A)}\psi$, $\langle\langle A\rangle\rangle_{E(A)}\psi$, nor

$\langle\langle A\rangle\rangle_{C(A)}\psi$. This is of course possible, because $\mathbb{E}_A$ (similarly to E_A) is not a KD45 modality (see Theorem 5 in Section 2.6.4). ■

Note that the semantics of CSL is based on *exactly the same* class of models as ATEL, ATOL, ATL$_{ir}$ etc. (i.e., on CEGS s). Thus, the above translation can also be used for reduction of validity (resp. satisfiability) problems for ATL$_{ir}$, ATOL and "Feasible ATEL" to weak validity (resp. satisfiability) of CSL. By Theorem 1, we have the following.

Corollary 2 *ATL$_{ir}$, ATOL and "Feasible ATEL" can be embedded in CSL.*

2.4.3 Constructive Strategic Logic *vs.* ATEL

As we already pointed out in Section 2.2.3, ATEL only enables expressing ability of type (4): the ATEL formula $\langle\langle A\rangle\rangle\varphi$ says that agents A may *happen* to behave in such a way that φ is enforced (but there might be no executable strategy to enforce it). Thus, ATEL is about a kind of ability different from the "constructive" one we study in this paper. Formally, ATEL differs from CSL in two main ways. First, it does not require uniform strategies. Second, it does not have the constructive knowledge operators.

First, consider non-uniformity. Note that uniform strategies is not a new idea of CSL (see Section 2.2), and that the differences between the ATEL operators and the uniform variants used by CSL are also shared by all the previously studied logics using uniform strategies. We nevertheless comment briefly on the difference here. First, the "nexttime" fragment of ATEL can be embedded in CSL, as the following proposition shows. It should be remembered that the CEGS s used in ATEL are slightly more general than the ones used in CSL (and the other approaches we have discussed): they do not require that the same actions are available in indistinguishable states. Below we refer to such CEGS s as *uniform CEGS s*.

Proposition 8 *Let φ be an ATEL formula that does not include operators $\Box$, $\mathcal{U}$ and M be a uniform CEGS. Then, $M, q \models_{\text{ATEL}} \varphi$ iff $M, q \models_{\text{CSL}} \varphi$.*

Proof It is sufficient to note that $M, q \models_{\text{ATEL}} \langle\langle A\rangle\rangle\bigcirc\varphi$ iff $M, q \models_{\text{CSL}} \langle\langle A\rangle\rangle\bigcirc\varphi$. Thus, we have that the "nexttime" formulae have the same semantics in both logics when interpreted at single states, and ATEL formulae include no "constructive" operators for aggregating sets of states. ■

Remark 6 *It is well known that cooperation modalities for strategies of perfect information (e.g., the ones in ATL and ATEL) have the following fixpoint characterizations:*

$$\langle\langle A\rangle\rangle\Box\varphi \quad\leftrightarrow\quad \varphi \wedge \langle\langle A\rangle\rangle\bigcirc\langle\langle A\rangle\rangle\Box\varphi, \tag{2.1}$$

$$\langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\psi \quad\leftrightarrow\quad \psi \vee \varphi \wedge \langle\langle A\rangle\rangle\bigcirc\langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\psi. \tag{2.2}$$

For uniform information strategies, the above formulae are not valid any more (see below). Still, it would be possible to embed the whole ATEL in CSL if we included fixpoint operators in the latter. In that case, the following translation could be used to translate ATL/ATEL modalities to equivalent CSL counterparts:

$$\begin{aligned} tr(\langle\langle A\rangle\rangle\Box\varphi) &= \nu\mathbf{Z}.\varphi \wedge \langle\langle A\rangle\rangle\bigcirc\mathbf{Z}, \\ tr(\langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\psi) &= \mu\mathbf{Z}.\psi \vee \varphi \wedge \langle\langle A\rangle\rangle\bigcirc\mathbf{Z}. \end{aligned}$$

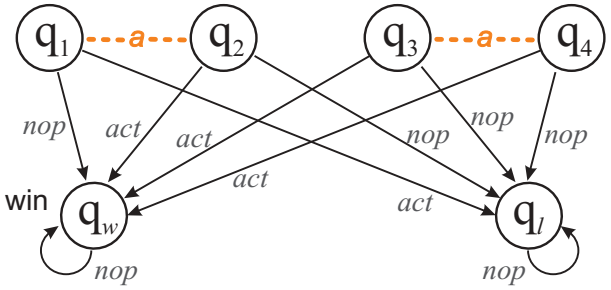


Figure 2.4: A model with one agent. From each of the states q_1, q_2, q_3, q_4, the same outcomes can be achieved in one step, albeit through different actions

We note that due to the uniformity of CSL strategies, the set of ATEL validities is not contained in CSL validities. A counter-example is the formula $\langle\langle r\rangle\rangle\Box\neg\mathsf{jail} \leftrightarrow \neg\mathsf{jail} \wedge \langle\langle r\rangle\rangle\bigcirc\langle\langle r\rangle\rangle\Box\neg\mathsf{jail}$. It is valid in ATEL (it is an instance of the valid scheme that gives a characterization of "always" in terms of "next" in ATL and ATEL). Still, the formula is false in model M_2 and state q_0 from Example 5: the left hand side of the biconditional is false, but the right hand side is true in M_2, q_0.

More importantly, we can show that CSL is more powerful than ATEL when we want to characterize sets of situations in actual systems. First, given a finite model, every ATEL formula has a CSL counterpart (i.e., a CSL formula which holds in exactly the same states). Second, CSL allows for finer-grained specifications than ATEL (in the sense that there are CSL formulae for which there are no ATEL formulae with the same extension). The result is formalized in Propositions 9 and 10.

Proposition 9 *Given a uniform CEGS, every ATEL formula has a CSL counterpart with the same extension (i.e., one which is satisfied in exactly the same states of the model).*

Proof sketch For finite models: let M be a model with $|M|$ states, and φ be an ATEL formula. All subformulae $\langle\langle A\rangle\rangle\Box\psi$ can be equivalently rewritten as $(\psi\wedge\langle\langle A\rangle\rangle\bigcirc)^{|M|}\psi$, where $|M|$ is the number of states in M. This follows by the property (2.1) above, and the fact that, after $|M|$ steps, the system is bound to come back to one of the previously visited states, for which a successful action has already been found. Similarly, subformulae $\langle\langle A\rangle\rangle\psi_1\,\mathcal{U}\,\psi_2$ can be equivalently rewritten as $(\psi_2 \vee \psi_1 \wedge \langle\langle A\rangle\rangle\bigcirc)^{|M|}\psi_2$. This way, we get an ATEL formula φ' without $\Box, \mathcal{U}$ which holds in exactly the same states as φ. By Proposition 8, φ' has the same extension in ATEL and CSL. ■

Proposition 10 *Given a uniform CEGS, there can be CSL formulae that have no ATEL counterpart with the same extension (i.e., one which is satisfied in exactly the same states of the model).*

Proof Consider model M_5 from Figure 2.4.3. The formula $\mathbb{K}_a\langle\langle a\rangle\rangle\bigcirc\mathsf{win}$ holds in q_3 and q_4, but not in q_1 nor q_2. There is no ATEL formula which is true exactly in q_3, q_4: it is easy to see that an ATEL formula is true in q_1 iff it is true in q_2 iff it is true in q_3 iff it is true in q_4. ■

2.4.4 Constructive Strategic Logic *vs.* ETSL

CSL and ETSL are underpinned by different notions of ability. ETSL can be treated as a logic that describes the outcome of *rational play* under imperfect information,[7] in the same way as CSL can be seen as a logic that captures agents' strategic abilities (regardless of whether the agents play rationally or not). Thus, the focus of CSL and ETSL is different, and we suspect that neither logic formally subsumes the other. However, several interesting associations have been already proposed in [68].

Let us consider only models with finite state spaces,[8] and formulae $\Phi \equiv \bigcirc\psi, \Box\psi$, or $\psi_1 \mathcal{U} \psi_2$ where ψ, ψ_1, ψ_2 are "vanilla" ETSL formulae.

Proposition 11 ([68]) *An agent has a strategy "de re" to enforce Φ if, and only if, he knows that his rational play will bring about Φ. Formally:*

$$M, q \models_{ETSL} K_a\langle\langle a\rangle\rangle\Phi \quad \textit{iff} \quad M, q \models_{CSL} \mathbb{K}_a\langle\langle a\rangle\rangle\Phi.$$

Proposition 12 ([68]) *If a coalition has common knowledge about how to play, then it has common knowledge that rational play will be successful:*

$$\textit{if} \quad M, q \models_{CSL} \mathbb{C}_A\langle\langle A\rangle\rangle\Phi \quad \textit{then} \quad M, q \models_{ETSL} C_A\langle\langle A\rangle\rangle\Phi.$$

The same holds for neither mutual nor distributed knowledge.

Proposition 13 ([68]) *If A have distributed knowledge that rational play will bring about Φ, then they have distributed knowledge how to play to bring about Φ. Formally:*

$$\textit{if} \quad M, q \models_{ETSL} D_A\langle\langle A\rangle\rangle\Phi \quad \textit{then} \quad M, q \models_{CSL} \mathbb{D}_A\langle\langle A\rangle\rangle\Phi.$$

The same holds for neither mutual nor common knowledge.

A more definitive study of this issue is beyond the scope of this paper.

2.5 Verification of Strategic Abilities through Model Checking

The *model checking* problem asks whether a given formula φ holds in a given model M and state q. We define the *general model checking* problem as the problem that asks whether formula φ holds in model M and *set of states* Q. Let $mctl(\varphi, M)$ be a CTL model checker that returns the set of all states which satisfy φ in M. Below, we sketch an algorithm $mcheck(\varphi, M, Q)$ that returns *true* if $M, Q \models_{CSL} \varphi$ and *false* otherwise, running in time $\mathbf{\Delta_2^P}$, i.e., in deterministic polynomial time with adaptive queries to an **NP** oracle.

Case $\varphi \equiv p$**:** return(*true*) if $p \in \pi(q)$ for all $q \in Q$, else return(*false*);

Case $\varphi \equiv \neg\psi$**:** return(*true*) if $mcheck(\psi, M, Q) =$ *false*, else return(*false*);

[7] We emphasize that this is a specific notion of rationality (i.e., agents are assumed to *play only undominated strategies*). Game theory proposes several other rationality criteria as well, based e.g. on Nash equilibrium, dominant strategies, or Pareto efficiency. In fact, it is easy to imagine ETSL-like logics based on these notions instead.

[8] More generally, we can consider models M such that there exists at least one undominated strategy wrt M, q, Φ.

Case $\varphi \equiv \psi_1 \wedge \psi_2$**:** return(*true*) if $mcheck(\psi_1, M, Q) =$ *true* and $mcheck(\psi_2, M, Q) =$ *true*, else return(*false*);

Case $\varphi \equiv \mathcal{K}_A\psi$**:** Compute $Q' := \mathrm{img}(Q, \sim_A^{\mathcal{K}})$, and then return(*true*) if $mcheck(\psi, M, q) =$ *true* for all $q \in Q'$, else return(*false*);

Case $\varphi \equiv \hat{\mathcal{K}}_A\psi$**:** return($mcheck(\psi, M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}))$);

Case $\varphi \equiv \langle\langle A\rangle\rangle \bigcirc \psi$**:** Run $mcheck(\psi, M, q)$ for every $q \in St$, and label the states in which the answer was *true* with an additional proposition yes (not used elsewhere). Then, guess the strategy of A, and "trim" model M by removing all the transitions inconsistent with the strategy (yielding a sparser model M'). Finally, return(*true*) if $Q \subseteq mctl(\mathsf{A}\bigcirc \mathsf{yes}, M')$, else return(*false*).

NOTE: subformula ψ is checked in the original model M, and not in M'!

Case $\varphi \equiv \langle\langle A\rangle\rangle \Box \psi$**:** Run $mcheck(\psi, M, q)$ for every $q \in St$, and label the states in which the answer was *true* with an additional proposition yes (not used elsewhere). Then, guess the strategy of A, and "trim" model M by removing all the transitions inconsistent with the strategy (yielding a sparser model M'). Finally, return(*true*) if $Q \subseteq mctl(\mathsf{A}\Box \mathsf{yes}, M')$, else return(*false*). Again, note that ψ is checked in the original model M.

Case $\varphi \equiv \langle\langle A\rangle\rangle \psi_1 \mathcal{U} \psi_2$**:** analogous.

As model checking CTL can be done in deterministic polynomial time [30], we get the following.

Proposition 14 *General model checking for Constructive Strategic Logic is in* $\mathbf{\Delta_2^P}$ *when the input size is measured with the number of transitions (and epistemic links) in the model, and the length of the formula.*

For the lower bound, we observe that CSL subsumes ATL$_{ir}$, and model checking ATL$_{ir}$ is $\mathbf{\Delta_2^P}$-complete [121, 81]. Thus, we pay no price in terms of complexity for using the more expressive language of CSL:

Theorem 2 *General model checking for Constructive Strategic Logic is* $\mathbf{\Delta_2^P}$*-complete in the number of transitions (and epistemic links) in the model, and the length of the formula.*

2.6 Constructive Knowledge

Philosophically, constructive knowledge draws inspiration from mathematical constructivism: in order to "constructively know" that φ, agents A must be able to find (or "construct") a mathematical object that supports φ. This is relevant when $\varphi \equiv \langle\langle B\rangle\rangle\psi$ – in that case, the mathematical object in question is a strategy for B which guarantees achieving ψ. The semantic role of *constructive knowledge operators* is to produce sets of states that will appear on the left hand side of the satisfaction relation. In a way, these modalities "aggregate" states into sets, and sets into bigger sets. On the other hand, most of

the other operators "split" (or "destroy") sets in the sense that, for evaluating $M, Q \models \varphi$, they require evaluation of subformulae of φ in single states rather than sets of states. Standard epistemic operators (C_A, E_A, D_A) are the most straightforward examples (e.g., evaluating $C_A\psi$ in M, Q "splits" into evaluating ψ in each state from $\mathrm{img}(Q, \sim_A^C)$ separately). Cooperation modalities (combined with temporal operators) are "splitting" in a similar way. Besides the "aggregating" and "splitting" operators, there are also "neutral" ones that do not change the set of reference: namely, conjunction ($\wedge$) and negation ($\neg$). In what follows, we study important properties of these operators in CSL.

2.6.1 Properties of Constructive Knowledge

In the following proposition we list some properties of constructive knowledge (keep in mind that strong validity implies validity).

Proposition 15 *The following are strongly valid for any $\hat{\mathcal{K}} \in \{\mathbb{C}, \mathbb{D}, \mathbb{E}\}$:*

1. $\hat{\mathcal{K}}_A(\varphi_1 \vee \varphi_2) \leftrightarrow (\hat{\mathcal{K}}_A\varphi_1 \vee \hat{\mathcal{K}}_A\varphi_2)$
2. $\hat{\mathcal{K}}_A\neg\varphi \leftrightarrow \neg\hat{\mathcal{K}}_A\varphi$
3. $\hat{\mathcal{K}}_A(\varphi_1 \wedge \varphi_2) \leftrightarrow (\hat{\mathcal{K}}_A\varphi_1 \wedge \hat{\mathcal{K}}_A\varphi_2)$
4. $\hat{\mathcal{K}}_A(\varphi_1 \rightarrow \varphi_2) \leftrightarrow (\hat{\mathcal{K}}_A\varphi_1 \rightarrow \hat{\mathcal{K}}_A\varphi_2)$

Proof

1. $M, Q \models \hat{\mathcal{K}}_A(\varphi_1 \vee \varphi_2)$ iff $M, \mathrm{img}(Q, \sim_A^{\hat{\mathcal{K}}}) \models \varphi_1 \vee \varphi_2$ iff $M, \mathrm{img}(Q, \sim_A^{\hat{\mathcal{K}}}) \models \varphi_1$ or $M, \mathrm{img}(Q, \sim_A^{\hat{\mathcal{K}}}) \models \varphi_2$ iff $M, Q \models \hat{\mathcal{K}}_A\varphi_1$ or $M, Q \models \hat{\mathcal{K}}_A\varphi_2$ iff $M, Q \models \hat{\mathcal{K}}_A\varphi_1 \vee \hat{\mathcal{K}}_A\varphi_2$.
2. $M, Q \models \hat{\mathcal{K}}_A\neg\varphi$ iff $M, \mathrm{img}(Q, \sim_A^{\hat{\mathcal{K}}}) \models \neg\varphi$ iff $M, \mathrm{img}(Q, \sim_A^{\hat{\mathcal{K}}}) \not\models \varphi$ iff $M, Q \not\models \hat{\mathcal{K}}_A\varphi$ iff $M, Q \models \neg\hat{\mathcal{K}}_A\varphi$.
3. $M, Q \models \hat{\mathcal{K}}_A(\varphi_1 \wedge \varphi_2)$ iff $M, \mathrm{img}(Q, \sim_A^{\hat{\mathcal{K}}}) \models \varphi_1 \wedge \varphi_2$ iff $M, \mathrm{img}(Q, \sim_A^{\hat{\mathcal{K}}}) \models \varphi_1$ and $M, \mathrm{img}(Q, \sim_A^{\hat{\mathcal{K}}}) \models \varphi_2$ iff $M, Q \models \hat{\mathcal{K}}_A\varphi_1$ and $M, Q \models \hat{\mathcal{K}}_A\varphi_2$ iff $M, Q \models \hat{\mathcal{K}}_A\varphi_1 \wedge \hat{\mathcal{K}}_A\varphi_2$.
4. $M, Q \models \hat{\mathcal{K}}_A(\neg\varphi_1 \vee \varphi_2)$ iff $M, Q \models (\hat{\mathcal{K}}_A\neg\varphi_1) \vee \hat{\mathcal{K}}_A\varphi_2$ iff $M, Q \models (\neg\hat{\mathcal{K}}_A\varphi_1) \vee \hat{\mathcal{K}}_A\varphi_2$ iff $M, Q \models \hat{\mathcal{K}}_A\varphi_1 \rightarrow \hat{\mathcal{K}}_A\varphi_2$.

■

2.6.2 Is $\mathbb{K}_a$ an Epistemic Operator?

We believe that operators $\mathbb{C}_A$, $\mathbb{E}_A$, $\mathbb{D}_A$ and $\mathbb{K}_a$ do capture a special kind of knowledge of agents. An interesting question is: do this notion of knowledge have the properties usually associated with knowledge? In particular, do postulates $\mathbf{K}, \mathbf{D}, \mathbf{T}, \mathbf{4}, \mathbf{5}$ of epistemic logic hold for constructive knowledge? In general, the answer is *no*; particularly, the truth axiom does not hold.

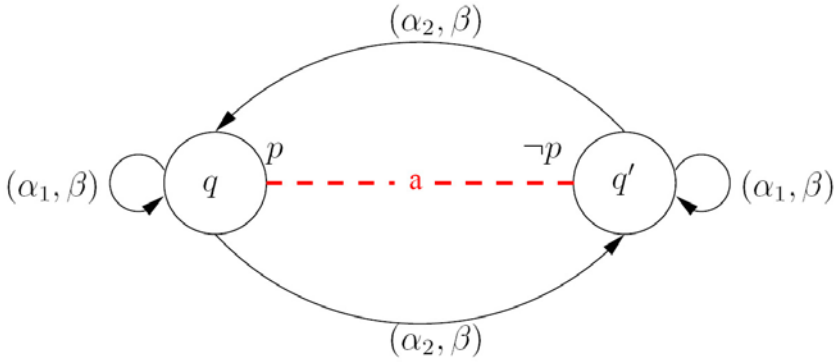


Figure 2.5: Model M_6 with two agents a, b, and two states q, q' such that $q \sim_a q'$

Theorem 3 *Below, we list the constructive knowledge versions of some of the S5 properties for individual agents. "Yes" means that the schema is strongly valid; "No" means that it is not even weakly valid (incidentally, none of the properties turns out to be weakly but not strongly valid).*

K	$\mathbb{K}_a(\varphi \to \psi) \to (\mathbb{K}_a\varphi \to \mathbb{K}_a\psi)$	*Yes*
D	$\neg\mathbb{K}_a\bot$	*Yes*
T	$\mathbb{K}_a\varphi \to \varphi$	*No*
4	$\mathbb{K}_a\varphi \to \mathbb{K}_a\mathbb{K}_a\varphi$	*Yes*
$\mathbf{4}^+$	$\mathbb{K}_a\varphi \leftrightarrow \mathbb{K}_a\mathbb{K}_a\varphi$	*Yes*
5	$\neg\mathbb{K}_a\varphi \to \mathbb{K}_a\neg\mathbb{K}_a\varphi$	*Yes*
$\mathbf{5}^+$	$\neg\mathbb{K}_a\varphi \leftrightarrow \mathbb{K}_a\neg\mathbb{K}_a\varphi$	*Yes*
B	$\varphi \to \mathbb{K}_a\neg\mathbb{K}_a\neg\varphi$	*No*

Before proving Theorem 3, we take a closer look at the relationship between satisfaction by a set of states ($M, Q \models \varphi$), and satisfaction in each of the states ($\forall_{q\in Q} M, q \models \varphi$). The following Lemma shows that the former does not necessarily imply the latter, and that the latter does not necessarily imply the former.

Lemma 1

1. *There is a model M, state q, agent a and formula φ such that $M, \mathrm{img}(q, \sim_a) \not\models \varphi$ and for every $q \in \mathrm{img}(q, \sim_a)$, $M, q \models \varphi$.*

2. *There are M, q, a, φ such that $M, \mathrm{img}(q, \sim_a) \models \varphi$ and $M, q \not\models \varphi$.*

Proof Consider model M_6 from Figure 2.6.2.

1) Let $\varphi = \langle\langle a\rangle\rangle \bigcirc p$. Now $M_6, q \models \varphi$ (a can choose action α_1), and $M_6, q' \models \varphi$ (a can choose action α_2). However, $M_6, \mathrm{img}(q, \sim_a) \not\models \varphi$, because no uniform strategy for a leads to q (in one step) from both q, q'.

2) Let $\varphi = \neg p$. Now $p \notin \pi(q) \cap \pi(q')$, so $M_6, \{q, q'\} \not\models p$, and $M_6, \mathrm{img}(q, \sim_a) \models \varphi$. But $p \in \pi(q)$, so $M_6, q \models p$, and $M_6, q \not\models \varphi$.

■

Proof of Theorem 3

K: Immediate by Proposition 15.

D: Suppose that $M, Q \models \mathbb{K}_a \bot$ for any $Q \neq \emptyset$. Then $M, \mathrm{img}(Q, \sim_a)) \models \bot$. By reflexivity of $\sim_a$, set $\mathrm{img}(Q, \sim_a)$ is nonempty, which contradicts Proposition 3.1.

T: Let M, q, a, φ be as in Lemma 1.2. $M, q \models \mathbb{K}_a\varphi$, but $M, q \not\models \varphi$, so **T** is not weakly (and hence not strongly) valid.

4^+/4: $M, Q \models \mathbb{K}_a\mathbb{K}_a\varphi$ iff $M, \mathrm{img}(Q, \sim_a) \models \mathbb{K}_a\varphi$ iff $M, \mathrm{img}(\mathrm{img}(Q, \sim_a), \sim_a) \models \varphi$ iff $M, \mathrm{img}(Q, \sim_a) \models \varphi$ (since $\mathrm{img}(\mathrm{img}(Q, \sim_a), \sim_a) = \mathrm{img}(Q, \sim_a)$) iff $M, Q \models \mathbb{K}_a\varphi$.

5^+/5: $M, Q \models \neg\mathbb{K}_a\varphi$ iff $M, Q \not\models \mathbb{K}_a\varphi$ iff, by 4^+, $M, Q \not\models \mathbb{K}_a\mathbb{K}_a\varphi$ iff, by Proposition 15, $M, Q \models \mathbb{K}_a\neg\mathbb{K}_a\varphi$.

B: Let M, q, a, φ be as in Lemma 1.1. $M, \mathrm{img}(q, \sim_a) \not\models \varphi$, so $M, q \models \mathbb{K}_a\neg\varphi$. By 4^+, $M, q \models \mathbb{K}_a\mathbb{K}_a\neg\varphi$, so $M, q \not\models \neg\mathbb{K}_a\mathbb{K}_a\neg\varphi$, and by Proposition 15 $M, q \not\models \mathbb{K}_a\neg\mathbb{K}_a\neg\varphi$. But $M, q \models \varphi$. Thus, **B** is not weakly (nor strongly) valid.

■

2.6.3 In Quest for the Truth Axiom

We have just showed that, out of the S5 properties, axioms $\mathbf{K}, \mathbf{D}, \mathbf{4}, \mathbf{5}$ (but not **T**!) hold. However, it also turns out that if we slightly restrict the language, then the corresponding **T** axiom becomes strongly valid. Let CSL$^-$ be the subset of CSL in which, between every occurrence of constructive knowledge ($\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$) and negation, there is always at least one operator other than conjunction.[9] Formally, CSL$^-$ formulae are defined by the following grammar (where $\mathcal{K} = C, E, D$ and $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$):

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle\bigcirc\varphi \mid \langle\langle A\rangle\rangle\Box\varphi \mid \langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\varphi \mid \mathcal{K}_A\varphi \mid \hat{\mathcal{K}}_A\psi,$$
$$\psi ::= p \mid \langle\langle A\rangle\rangle\bigcirc\varphi \mid \langle\langle A\rangle\rangle\Box\varphi \mid \langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\varphi \mid \mathcal{K}_A\varphi \mid \psi \wedge \psi \mid \hat{\mathcal{K}}_A\psi.$$

Theorem 4 *Every CSL$^-$ instance of* **T** *(i.e., $\mathbb{K}_a\psi \to \psi$) is strongly valid.*

Proof in the Appendix.

Thus, the **T** axiom holds for CSL$^-$. Note that, by Proposition 15, the meaning of negation or conjunction in the immediate scope of a constructive knowledge operator is the same as if the operator were immediately outside the constructive knowledge operator.[10] In consequence, every formula of the full CSL is equivalent to one in CSL$^-$. Thus, we can restrict our logical language to CSL$^-$ without losing expressive power, and we automatically

[9] In particular, the requirement is met when operators $\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$ are never immediately followed by either $\neg$ or $\wedge$.

[10] Which is very much *unlike* the semantics of negation following a standard knowledge operator!

"get" axiom **T**. We also observe that, from a more philosophical perspective, it is hard to pinpoint the intuitive meaning of negation immediately following constructive knowledge. Note that, e.g., $\mathbb{K}_a\neg\langle\langle a\rangle\rangle\varphi$ should be read as "a has *constructive* knowledge about being *unable* to achieve φ".[11] It seems thus, first, that the weaker version of the truth axiom in Theorem 4 might be more appropriate for constructive knowledge, and second, that it might be a good idea to consider the logical language of constructive knowledge to be limited to CSL$^-$. In this case, constructive knowledge has the **T** property, we do not lose any expressive power, and we leave out only formulae with philosophically unclear reading.

Is then the constructive knowledge in CSL$^-$ S5? First, it must be noted that – even though CSL and CSL$^-$ are expressively equivalent – the extension of the schema **T** is *different* in CSL$^-$ (for example, $\mathbb{K}_a\neg p \rightarrow \neg p$ is a CSL instance of **T**, but even though it is equivalent to the CSL$^-$ formula $\neg\mathbb{K}_a p \rightarrow \neg p$, the latter is *not* a CSL$^-$ instance of **T**). More importantly, in CSL$^-$ the axiom schemata **K** and **5**, at least written as in Theorem 3, are not valid, but they are not invalid either – they are simply not formulae at all. It does not seem correct to say that an operator has the S5 properties when it cannot even *express* the **K** principle or negative introspection. Furthermore, CSL$^-$ lacks the S5 principle of *uniform substitution*.

2.6.4 Properties of Collective Constructive Knowledge

We briefly consider the properties of collective knowledge operators. Theorem 5 should come as no surprise: note that, analogously to standard knowledge, constructive common and distributed knowledge have the same properties as individual knowledge, while mutual knowledge ("everybody knows") differs in that it does not satisfy the introspection axioms **4** and **5**.

Theorem 5 *Below, we list some of the S5 properties for collective constructive knowledge operators. We don't state the properties explicitly, but refer to Theorem 3 – axiom* **K** *for* $\mathbb{C}_A$ *becomes* $\mathbb{C}_A(\varphi \rightarrow \psi) \rightarrow (\mathbb{C}_A\varphi \rightarrow \mathbb{C}_A\psi)$*, and so on. "Yes" means that the schema is strongly valid; "No" means that it is not even weakly valid (the proof is left for the reader).*

	$\mathbb{C}_A$	$\mathbb{E}_A$	$\mathbb{D}_A$
K	*Yes*	*Yes*	*Yes*
D	*Yes*	*Yes*	*Yes*
T	*No*	*No*	*No*
4	*Yes*	*No*	*Yes*
$\mathbf{4}^+$	*Yes*	*No*	*Yes*
5	*Yes*	*No*	*Yes*
$\mathbf{5}^+$	*Yes*	*No*	*Yes*
B	*No*	*No*	*No*

Note that the proof of Theorem 4 required only that the epistemic relation in question was reflexive. Thus, it can be easily extended to handle collective constructive knowledge.

[11] $\mathbb{K}_a\langle\langle a\rangle\rangle\neg\varphi$, on the other hand, makes perfect sense: it refers to a's constructive *ability* to prevent φ.

Corollary 3 *Every CSL$^-$ instance of schema* **T** *for collective constructive knowledge operators $\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$ is strongly valid.*

2.7 Negation, Localization, and Definability of Knowledge

The semantics of negation presented in Section 2.3.1 (we call it *weak* negation from now on) yields a very strong notion of disjunction, as Proposition 3 states. Such a strong notion of disjunction makes sense when we talk about agents' abilities, i.e., when used inside a $\mathbb{K}_a$ operator. For example: $M, q \models \mathbb{K}_a(\langle\langle a\rangle\rangle\varphi \vee \langle\langle a\rangle\rangle\psi)$ means in fact that a in q can either identify a plan to achieve φ or to achieve ψ. On the other hand, for a disjunction of simpler formulae, e.g., primitive propositions p and r, a weaker notion seems more intuitive: the disjunction $p \vee r$ should hold in M, Q iff, for any state $q \in Q$, at least one of the disjuncts p and r holds in q (but different disjuncts may hold in different states of Q). This intuition can be captured with a different negation operator $\sim$, which we call "strong" negation. The idea of strong negation can be summarized as: $M, Q \models \sim\varphi$ iff $M, q \not\models \varphi$ for every $q \in Q$. However, we will define it in terms of another, more primitive operator that we call *localization*.

As it turns out, the significance of localization goes beyond our discussion on various kinds of negation. Most importantly, localization can be used to define standard knowledge operators from constructive knowledge operators. On the other hand, localization itself proves definable from strategic and temporal operators. In consequence, standard knowledge can be defined in CSL without standard knowledge operators.

2.7.1 Local Evaluation of Formulae

In the semantics of CSL, formulae are interpreted in sets of states; in order for φ to hold in M, Q, the formula must be "globally" satisfied in all states from Q at once (i.e., with single evidence). Another option is to evaluate φ *locally* in particular states from Q. To this end, we introduce a modality that specifies explicitly that the formula must be evaluated for every relevant state separately:

$$M, Q \models \mathsf{loc}\,\varphi \text{ iff } M, q \models \varphi \text{ for every } q \in Q.$$

Proposition 16 *Below, we investigate some typical axioms with respect to the localization modality. "Yes" means that the scheme is strongly valid, "No" means that the scheme is not strongly valid. Note that all the schemes below are weakly valid, because $M, q \models \mathsf{loc}\,\varphi \leftrightarrow \varphi$ for every individual state q.*

K	$\mathsf{loc}\,(\varphi \to \psi) \to (\mathsf{loc}\,\varphi \to \mathsf{loc}\,\psi)$	*Yes*
D	$\neg\mathsf{loc}\,\bot$	*Yes*
T	$\mathsf{loc}\,\varphi \to \varphi$	*No*
4	$\mathsf{loc}\,\varphi \to \mathsf{loc}\,\mathsf{loc}\,\varphi$	*Yes*
$\mathbf{4}^+$	$\mathsf{loc}\,\varphi \leftrightarrow \mathsf{loc}\,\mathsf{loc}\,\varphi$	*Yes*
5	$\neg\mathsf{loc}\,\varphi \to \mathsf{loc}\,\neg\mathsf{loc}\,\varphi$	*No*
$\mathbf{5}^+$	$\neg\mathsf{loc}\,\varphi \leftrightarrow \mathsf{loc}\,\neg\mathsf{loc}\,\varphi$	*No*
B	$\varphi \to \mathsf{loc}\,\neg\mathsf{loc}\,\neg\varphi$	*Yes*

Proof in the Appendix.

Thus, localization is weak, but not strong, S5. In particular, S5 properties **T** and **5** do not necessarily hold in some contexts, for example in the immediate scope of a constructive knowledge operator.

Proposition 17 *Some other localization properties are the following, all strongly valid (proof is left for the reader).*

$\mathsf{loc}\,p \leftrightarrow p,\ p \in \Pi$ $\qquad$ $\mathsf{loc}\,(\varphi \wedge \psi) \leftrightarrow (\mathsf{loc}\,\varphi \wedge \mathsf{loc}\,\psi)$

$\langle\langle A\rangle\rangle \bigcirc \varphi \leftrightarrow \langle\langle A\rangle\rangle \bigcirc \mathsf{loc}\,\varphi$ $\qquad$ $\langle\langle A\rangle\rangle \Box \varphi \leftrightarrow \langle\langle A\rangle\rangle \Box \mathsf{loc}\,\varphi$

$\langle\langle A\rangle\rangle \varphi\, \mathcal{U}\, \psi \leftrightarrow \langle\langle A\rangle\rangle \mathsf{loc}\,\varphi\, \mathcal{U}\, \psi \leftrightarrow \langle\langle A\rangle\rangle \varphi\, \mathcal{U}\, \mathsf{loc}\,\psi$

$\mathsf{loc}\,\mathcal{K}_A\varphi \leftrightarrow \mathcal{K}_A\varphi$ $\qquad$ $\mathcal{K}_A\varphi \leftrightarrow \mathcal{K}_A\mathsf{loc}\,\varphi,\ \mathcal{K} \in \{C, E, D\}$

We will show in the following sections how the loc operator can be used to define standard knowledge and alternative negation operators. This makes the following result very important: it says that localization is definable in the CSL language, from the $\langle\langle\emptyset\rangle\rangle$ and $\mathcal{U}$ operators.

Proposition 18 *The following formula is strongly valid:*

$$\mathsf{loc}\,\varphi \leftrightarrow \langle\langle\emptyset\rangle\rangle \varphi\, \mathcal{U}\, \varphi$$

Proof $M, Q \models \langle\langle\emptyset\rangle\rangle \varphi\, \mathcal{U}\, \varphi$ iff $\forall_{\lambda \in out(Q,\emptyset)}$ there is an $i \geq 0$ such that $M, \lambda[i] \models \varphi$ and for any j such that $0 \leq j < i$, $M, \lambda[j] \models \varphi$. Since for each $q \in Q$ there is a $\lambda \in out(Q, \emptyset)$ with $\lambda[0] = q$, this implies that $\forall_{q \in Q} M, q \models \varphi$ which is the same as $M, Q \models \mathsf{loc}\,\varphi$. To see that the other direction holds as well, assume that $M, Q \models \mathsf{loc}\,\varphi$ and let $\lambda \in out(Q, \emptyset)$. We must provide a witness for i; take $i = 0$. Now, $M, \lambda[i] \models \varphi$ and there is no j such that $0 \leq j < i$, so $M, Q \models \langle\langle\emptyset\rangle\rangle \varphi\, \mathcal{U}\, \varphi$. ■

2.7.2 Defining Standard Knowledge from Constructive Knowledge

Standard knowledge operators are definable from constructive knowledge and localization:

Proposition 19 $\mathcal{K}_A\varphi \leftrightarrow \hat{\mathcal{K}}_A\mathsf{loc}\,\varphi$ *is strongly valid for any* $\mathcal{K} \in \{C, E, D\}$, $\hat{C} = \mathbb{C}$, $\hat{E} = \mathbb{E}$, $\hat{D} = \mathbb{D}$.

Proof $M, Q \models \hat{\mathcal{K}}_A\mathsf{loc}\,\varphi$ iff $M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}) \models \mathsf{loc}\,\varphi$ iff $\forall_{q \in \mathrm{img}(Q, \sim_A^{\mathcal{K}})} M, q \models \varphi$ iff $M, Q \models \mathcal{K}_A\varphi$. ■

In particular, knowledge of a formula is the same as constructive knowledge of the localization of the formula, i.e. $K_a\varphi \leftrightarrow \mathbb{K}_a\mathsf{loc}\,\varphi$. An important corollary of Propositions 19 and 18 is the following.

Theorem 6 *The following is strongly valid:*

$$\mathcal{K}_A\varphi \leftrightarrow \hat{\mathcal{K}}_A\langle\langle\emptyset\rangle\rangle\varphi\,\mathcal{U}\,\varphi.$$

Theorem 6 shows that *standard knowledge can be seen as a special case of constructive knowledge*. It follows that the standard knowledge operators are strictly speaking redundant in the CSL language.

2.7.3 Non-Standard Definitions of Negation

Negation, as defined in Section 2.3.1, is "weak" in the sense that it is sufficient for the negation of, e.g., an atomic formula p to hold in a set of states Q that p is false in at least one state from Q. Several other interpretations of negation in a set of states are possible, corresponding to different ways of quantifying over the set. We define *strong negation* as:

$$\sim\varphi \equiv \mathsf{loc}\,\neg\varphi$$

Note that, by Proposition 18, strong negation is definable from weak negation: $\sim\varphi$ can be equivalently defined as $\langle\langle\emptyset\rangle\rangle(\neg\varphi)\,\mathcal{U}\,(\neg\varphi)$.

Proposition 20 $M, Q \models \sim\varphi$ *iff, for every* $q \in Q$*, we have that* $M, q \not\models \varphi$*.*

Proof $M, Q \models \sim\varphi$ iff $M, Q \models \mathsf{loc}\,\neg\varphi$ iff for every $q \in Q$ we have that $M, q \not\models \varphi$. ■

Strong negation does *not* behave as classical negation: it does not obey the law of double negation, the law of excluded middle, or the consistency requirement under *strong* validity. Nevertheless, it preserves these laws under weak validity.

Proposition 21

1. $\sim\sim\varphi \rightarrow \varphi$ *is weakly valid, but not strongly valid.*
2. $\varphi \rightarrow \sim\sim\varphi$ *is weakly valid, but not strongly valid.*
3. $\varphi \vee \sim\varphi$ *is weakly valid, but not strongly valid.*
4. $\neg(\varphi \wedge \sim\varphi)$ *is weakly valid, but not strongly valid.*

Proof

1), 2) Weak validity is immediate. $M, Q \models \sim\sim\varphi$ iff $M, Q \models \mathsf{loc}\,\neg\,\sim\varphi$ iff $M, Q \models \mathsf{loc}\,\neg\mathsf{loc}\,\neg\varphi$ iff for every $q \in Q$ we have that $M, q \models \varphi$. Counter-examples for the two implications are found in the two parts of Lemma 1, respectively, by taking $Q = \mathrm{img}(q, \sim_a)$.

3) Weak validity is immediate. As a counter-example to strong validity, take M and φ from Lemma 1.1, and let $Q = \mathrm{img}(q, \sim_a)$. $M, Q \not\models \varphi$, and it is not the case that $M, q' \not\models \varphi$ for every $q' \in Q$.

4) Weak validity: immediate. Strong validity: take $M = M_6$ from Lemma 1, and let $Q = \{q, q'\}$, $\varphi \equiv \neg\langle\langle a\rangle\rangle\bigcirc p$.

■

Remark 7 *Alternatively, strong negation can be taken as a primary notion: localization is definable from strong negation, and standard knowledge is thus definable from constructive knowledge and strong negation. Formally, the following are strongly valid:*

1. $\mathsf{loc}\,\varphi \leftrightarrow {\sim}{\sim}\varphi$
2. $\mathcal{K}_A\varphi \leftrightarrow \hat{\mathcal{K}}_A\ {\sim}{\sim}\varphi$.

Boolean Operators Based on Strong Negation

Recall that connectives like $\vee$ and $\rightarrow$ are defined in terms of weak negation ($\neg$). Similar connectives can be defined for strong negation:

- $\varphi_1 \parallel \varphi_2 \equiv {\sim}({\sim}\varphi_1 \wedge {\sim}\varphi_2)$,
- $\varphi_1 \rightsquigarrow \varphi_2 \equiv {\sim}\varphi_1 \parallel \varphi_2$, and
- $\varphi_1 \leftrightsquigarrow \varphi_2 \equiv (\varphi_1 \rightsquigarrow \varphi_2) \wedge \varphi_2 \rightsquigarrow \varphi_1$.

These versions of disjunction, material implication, and material biconditional have the following semantic characterizations:

Proposition 22

1. $M, Q \models \varphi_1 \parallel \varphi_2$ *iff, for every* $q \in Q$, *we have* $M, q \models \varphi_1$ *or* $M, q \models \varphi_2$;
2. $M, Q \models \varphi_1 \rightsquigarrow \varphi_2$ *iff, for every* $q \in Q$, *we have that* $M, q \models \varphi_1$ *implies* $M, q \models \varphi_2$;
3. $M, Q \models \varphi_1 \leftrightsquigarrow \varphi_2$ *iff, for every* $q \in Q$, *we have that* $M, q \models \varphi_1$ *iff* $M, q \models \varphi_2$.

Proof

1. $M, Q \models {\sim}({\sim}\varphi_1 \wedge {\sim}\varphi_2)$ iff $\forall_{q\in Q} M, q \not\models {\sim}\varphi_1 \wedge {\sim}\varphi_2$ iff $\forall_{q\in Q} M, q \not\models {\sim}\varphi_1$ or $M, q \not\models {\sim}\varphi_2$ iff $\forall_{q\in Q} M, q \models \varphi_1$ or $M, q \models \varphi_2$.
2. $M, Q \models \varphi_1 \rightsquigarrow \varphi_2$ iff $M, Q \models {\sim}\varphi_1 \parallel \varphi_2$ iff $\forall_{q\in Q}(M, q \models {\sim}\varphi_1$ or $M, q \models \varphi_2)$ iff $\forall_{q\in Q}(M, q \not\models \varphi_1$ or $M, q \models \varphi_2)$ iff $\forall_{q\in Q} M, q \models \varphi_1 \rightarrow \varphi_2$.
3. Straightforward.

■

We can also define the strong negation-based versions of Boolean constants "true" and "false", but they coincide with the ones already proposed in Section 2.3.2.

Proposition 23 *Let* $\bot \equiv p \wedge {\sim}p$, *and* $\top \equiv {\sim}\bot$. *Then:*

1. $M, Q \not\models \bot$ *for all* $Q \subseteq St$, $Q \neq \emptyset$.
2. $M, Q \models \top$ *for all* $Q \subseteq St$, $Q \neq \emptyset$.

Proof Straightforward. ■

Some Connections between the Weak and the Strong

It is immediate from Proposition 22 that, just as strong negation is the localization of weak negation, the operators $\|$, $\rightsquigarrow$ and $\leftrightsquigarrow$ defined by strong negation, are the localizations of their counterparts $\vee$, $\rightarrow$, $\leftrightarrow$ defined by weak negation:

Proposition 24 *The following are strongly valid:*

$$(\varphi_1 \parallel \varphi_2) \leftrightarrow \mathsf{loc}\,(\varphi_1 \vee \varphi_2)$$
$$(\varphi_1 \rightsquigarrow \varphi_2) \leftrightarrow \mathsf{loc}\,(\varphi_1 \rightarrow \varphi_2)$$
$$(\varphi_1 \leftrightsquigarrow \varphi_2) \leftrightarrow \mathsf{loc}\,(\varphi_1 \leftrightarrow \varphi_2)$$

Moreover, for validity (not strong validity), the two negations, the two disjunctions and the two implications coincide:

Proposition 25 *The following formulae are valid (but not strongly valid):*

1. $\neg\varphi \leftrightarrow \sim\varphi$
2. $(\varphi_1 \vee \varphi_2) \leftrightarrow (\varphi_1 \parallel \varphi_2)$
3. $(\varphi_1 \rightarrow \varphi_2) \leftrightarrow (\varphi_1 \rightsquigarrow \varphi_2)$

Proof Immediate from Proposition 24, since $M, q \models \psi$ iff $M, q \models \mathsf{loc}\,\psi$, for any (single) state q. ■

The following proposition shows that the notions of strong and weak validity can be seen as dual with respect to the strong and weak versions of the connectives.

Proposition 26 1. *$\sim\varphi$ is strongly valid iff $\neg\varphi$ is weakly valid.*

2. *$\varphi_1 \parallel \varphi_2$ is strongly valid iff $\varphi_1 \vee \varphi_2$ is weakly valid.*
3. *$\varphi_1 \rightsquigarrow \varphi_2$ is strongly valid iff $\varphi_1 \rightarrow \varphi_2$ is weakly valid.*
4. *$\varphi_1 \leftrightsquigarrow \varphi_2$ is strongly valid iff $\varphi_1 \leftrightarrow \varphi_2$ is weakly valid.*

The laws of negation were stated in Proposition 21 using connectives $\vee$, etc., defined from weak negation. We can now show, however, that the laws of negation do in fact hold for strong negation if we state these laws using the operators defined from strong negation.

Proposition 27 1. *$\sim\sim\varphi \leftrightsquigarrow \varphi$ is strongly valid.*

2. *$\varphi \parallel \sim\varphi$ is strongly valid.*
3. *$\sim(\varphi \wedge \sim\varphi)$ is strongly valid.*

Proof Immediate from Propositions 21 and 26. ■

Properties of Constructive Knowledge with "Strong" Negation

In Section 2.6.2, we discussed the S5 properties of constructive knowledge. These properties can also be stated using strong negation, and derived connectives, instead of weak negation.

Theorem 7 *Below, we list constructive knowledge versions of some S5 properties using strong negation. "Yes" means that the schema is strongly valid; "No" means that it is not even weakly valid (again, none of the properties turn out to be weakly but not strongly valid).*

$\partial\mathbf{K}$	$\mathbb{K}_a(\varphi \rightsquigarrow \psi) \rightsquigarrow (\mathbb{K}_a\varphi \rightsquigarrow \mathbb{K}_a\psi)$	*No*
$\partial\mathbf{D}$	$\sim\mathbb{K}_a \underline{\lambda}$	*Yes*
$\partial\mathbf{T}$	$\mathbb{K}_a\varphi \rightsquigarrow \varphi$	*No*
$\partial\mathbf{4}$	$\mathbb{K}_a\varphi \rightsquigarrow \mathbb{K}_a\mathbb{K}_a\varphi$	*Yes*
$\partial\mathbf{4}^+$	$\mathbb{K}_a\varphi \leftrightsquigarrow \mathbb{K}_a\mathbb{K}_a\varphi$	*Yes*
$\partial\mathbf{5}$	$\sim\mathbb{K}_a\varphi \rightsquigarrow \mathbb{K}_a\sim\mathbb{K}_a\varphi$	*Yes*
$\partial\mathbf{5}^+$	$\sim\mathbb{K}_a\varphi \leftrightsquigarrow \mathbb{K}_a\sim\mathbb{K}_a\varphi$	*Yes*
$\partial\mathbf{B}$	$\varphi \rightsquigarrow \mathbb{K}_a\sim\mathbb{K}_a\sim\varphi$	*Yes*

Proof in the Appendix.

Finally, we point out that if we restrict the language to CSL$^-$, as discussed in Section 2.6.3, we get the truth axiom $\partial\mathbf{T}$, i.e., the following variant of Theorem 4.

Theorem 8 *Every CSL$^-$ instance of schema $\partial\mathbf{T}$ ($\mathbb{K}_a\varphi \rightsquigarrow \varphi$) is strongly valid.*

Proof Note that $\forall_{q\in Q} M, q \models \mathbb{K}_a\varphi \rightarrow \varphi$ (by $\mathbf{T}$), which implies that $M, Q \models \mathbb{K}_a\varphi \rightsquigarrow \varphi$ (by Proposition 22). ■

Other Negations

We have considered two operators for negation so far. Yet another alternative is: $\angle\varphi \equiv \neg\mathsf{loc}\,\varphi$. The meaning of $\angle$ is characterized with the following proposition.

Proposition 28 *$M, Q \models \angle\varphi$ iff there exists $q \in Q$ such that $M, q \not\models \varphi$.*

Proof $M, Q \models \angle\varphi$ iff $M, Q \models \neg\mathsf{loc}\,\varphi$ iff $M, Q \not\models \mathsf{loc}\,\varphi$ iff there is a $q \in Q$ such that $M, q \not\models \varphi$. ■

2.8 Normal Forms and Expressiveness

In this section, we investigate expressiveness further, with particular focus on the relationship between localization, weak negation and strong negation. In order to study expressiveness, we will study variants of the language defined in Section 2.3.1 with other (primary) operators. We have discussed the interpretation of the following operators in sets of states:

$$\neg \quad \wedge \quad \langle\langle A\rangle\rangle T \quad C_A \quad E_A \quad D_A \quad \mathbb{C}_A \quad \mathbb{E}_A \quad \mathbb{D}_A \quad \mathsf{loc} \quad \sim$$

where T is an ATL temporal connective and A is a set of agents. We use the expression $\mathcal{L}(\neg, \wedge, \langle\langle A\rangle\rangle T, \mathcal{K}_A, \hat{\mathcal{K}}_A, \mathsf{loc}\,, \sim)$ to denote the language with all the mentioned operators, $\mathcal{L}(\neg, \wedge, \langle\langle A\rangle\rangle T, \mathcal{K}_A, \hat{\mathcal{K}}_A, \mathsf{loc}\,)$ to denote the language with all operators except strong negation, and so on. The CSL language introduced in Section 2.3.1 is $\mathcal{L} = \mathcal{L}(\neg, \wedge, \langle\langle A\rangle\rangle T, \mathcal{K}_A, \hat{\mathcal{K}}_A)$. For simplicity, we sometimes use $\mathcal{L}^*$ for the most extensive language $\mathcal{L}(\neg, \wedge, \langle\langle A\rangle\rangle T, \mathcal{K}_A, \hat{\mathcal{K}}_A, \mathsf{loc}\,, \sim)$.

We say that two formulae φ and ψ are *equivalent*, if $\varphi \leftrightarrow \psi$ is valid, and that they are *strongly equivalent* if $\varphi \leftrightarrow \psi$ is strongly valid. We say that a language $\mathcal{L}_2$ is *at least as expressive* as a language $\mathcal{L}_1$, if for every $\varphi_1 \in \mathcal{L}_1$ there exist an equivalent $\varphi_2 \in \mathcal{L}_2$. We say that $\mathcal{L}_2$ and $\mathcal{L}_1$ are *expressively equivalent*, if $\mathcal{L}_2$ is at least as expressive as $\mathcal{L}_1$ and $\mathcal{L}_1$ is at least as expressive as $\mathcal{L}_2$.

We will make use of the following definition:

$$Atoms = \Theta \cup \{\langle\langle A\rangle\rangle T\gamma : \gamma \in \mathcal{L}^*\} \cup \{\sim\gamma : \gamma \in \mathcal{L}^*\}.$$

We begin with defining a *normal form* of our formulae.

2.8.1 Constructive Normal Form

A formula, possibly containing strong negation, is of *constructive normal form* if every subformula starting with a $\hat{\mathcal{K}}_A$ operator is of the form $\hat{\mathcal{K}}_1 \cdots \hat{\mathcal{K}}_k\psi$ where ψ is either a primitive proposition, starts with a cooperation modality, or starts with strong negation. We now show that every $\mathcal{L}^*$ formula is equivalent to one of constructive normal form, and also to a formula of constructive normal form without strong negation.

Definition 1 (Constructive normal form (CSNF)) *The set of $\mathcal{L}^*$ formulae of* constructive normal form (CSNF) *is defined inductively as follows.*

- *p is of CSNF when $p \in \Theta$,*
- *$\hat{\mathcal{K}}\gamma$ is of CSNF iff γ is of CSNF and either $\gamma \in Atoms$ or $\gamma = \hat{\mathcal{K}}'\chi$,*
- *$\langle\langle G\rangle\rangle T\gamma$ is of CSNF iff γ is of CSNF,*
- *$\neg\gamma$ is of CSNF iff γ is of CSNF,*
- *$\gamma_1 \wedge \gamma_2$ is of CSNF iff both γ_1 and γ_2 are of CSNF,*
- *$\sim\gamma$ is of CSNF iff γ is of CSNF.*

Theorem 9 *Every formula in $\mathcal{L}^*$ is strongly equivalent to a formula of constructive normal form.*

Proof in the Appendix.

Thus, any formula of the most general kind we have considered is equivalent to a formula of CSNF. Note that a CSNF formula might contain strong negation. However, we can also get rid of strong negation, as the following result states.

Corollary 4 *Every formula in $\mathcal{L}^*$ is strongly equivalent to a formula of constructive normal form without strong negation.*

Proof in the Appendix.

2.8.2 Expressiveness of Strong Negation

We have shown in Section 2.7 that standard knowledge, localization and strong negation can be defined with use of weak negation (together with conjunction, constructive knowledge and ATL operators). Thus, $\mathcal{L}(\neg, \wedge, \langle\langle A\rangle\rangle T, \hat{\mathcal{K}})$ is already as expressive as the full $\mathcal{L}^*$. Now we will investigate the other direction: does *weak* negation add expressiveness if we already have strong negation? We show in the following theorem that, in the language $\mathcal{L}$ extended with strong negation, every formula is actually equivalent to one without weak negation.

Theorem 10 *Every formula in $\mathcal{L}(\neg, \wedge, \langle\langle A\rangle\rangle T, \hat{\mathcal{K}}, \sim)$ is equivalent to a formula of $\mathcal{L}(\wedge, \langle\langle A\rangle\rangle T, \hat{\mathcal{K}}, \sim)$.*

Proof in the Appendix.

Thus, in particular, the following four languages are expressively equivalent:

$$\mathcal{L}(\neg, \wedge, \langle\langle A\rangle\rangle T, \hat{\mathcal{K}}) \quad \mathcal{L}(\wedge, \langle\langle A\rangle\rangle T, \hat{\mathcal{K}}, \sim) \qquad \mathcal{L} \qquad \mathcal{L}^*$$

In consequence, both $\mathcal{L}(\neg, \wedge, \langle\langle A\rangle\rangle T, \hat{\mathcal{K}})$ and $\mathcal{L}(\wedge, \langle\langle A\rangle\rangle T, \hat{\mathcal{K}}, \sim)$ are expressively complete with respect to the other operators we have considered. An important difference between $\mathcal{L}(\neg, \wedge, \langle\langle A\rangle\rangle T, \hat{\mathcal{K}})$ and $\mathcal{L}(\wedge, \langle\langle A\rangle\rangle T, \hat{\mathcal{K}}, \sim)$ is that strong negation is definable from weak negation and ATL operators by a simple schema $(\langle\langle\emptyset\rangle\rangle \neg\varphi \,\mathcal{U}\, \neg\varphi)$, while this is not the case when we reverse the roles of the negations.

2.9 Conclusions

In this paper, we propose a non-standard semantics for the modal logic of strategic ability under imperfect information, in which formulae are interpreted over *sets of states* rather than in single states.[12] Moreover, we introduce new epistemic operators for "constructive" knowledge. It turns out that, in this new semantics, simple cooperation modalities $\langle\langle A\rangle\rangle$ can be combined with "constructive" epistemic operators into sufficiently expressive formulae. Indeed, the new logic *is strictly more expressive than most existing ATL versions for imperfect information*, while it retains the same model checking complexity as the least costly of them. The philosophical dimension of constructive knowledge is also natural: the constructive knowledge operators capture the notion of *knowing "de re"*, while the standard epistemic operators refer to *knowing "de dicto"*. Moreover, it turns out that standard (traditional) knowledge is a special case of constructive knowledge. Also, the language of CSL is expressive enough to enable expressing several other interesting operators in a simple way.

Most of the usual S5 properties (with the notable exception of the truth axiom T) hold for constructive knowledge. Furthermore, if we slightly restrict the syntax of CSL, we do not lose expressive power and the schema T becomes a validity.

[12] We emphasize again that we do not propose new models: concurrent epistemic game structures have already been used for several years in ATEL-like logics. What we propose is a new interpretation of formulae.

CSL has novel, meaningful epistemic operators that can be used to capture important properties of the interaction between knowledge, action and ability. In future work, we plan to investigate further the expressivity of CSL, and its relationship with logics like ETSL, ATL$_{iR}$, ATEL-R*, ATEL-A, and "Uniform STIT". A good case study (together with a more detailed analysis of verification complexity) is essential to determine the applicability of the logic. Also, the (relative) expressive power of various operators in our semantics seems to be worth further study.

We thank anonymous reviewers of JANCL and AAMAS-06 for their helpful remarks. Thomas Ågotnes' work has been supported by the Research Council of Norway under grant 166525/V30. Wojtek Jamroga would also like to thank Jan Broersen, John-Jules Meyer and Wiebe van der Hoek.

2.10 Appendix: Some Proofs

Theorem 1

Proof (structural induction with respect to the structure of φ)

- $M,q \models_{\text{CSL}} tr(p)$ iff $M,q \models_{\text{CSL}} p$ iff $p \in \pi(q)$ iff $M,q \models_{\mathcal{L}} p$.
- $M,q \models_{\text{CSL}} tr(\neg\varphi)$ iff $M,q \not\models_{\text{CSL}} tr(\varphi)$ iff (by induction) $M,q \not\models_{\mathcal{L}} \varphi$ iff $M,q \models_{\mathcal{L}} \neg\varphi$.
- $M,q \models_{\text{CSL}} tr(\varphi \wedge \psi)$ iff $M,q \models_{\text{CSL}} tr(\varphi)$ and $M,q \models_{\text{CSL}} tr(\psi)$ iff (by induction) $M,q \models_{\mathcal{L}} \varphi$ and $M,q \models_{\mathcal{L}} \psi$ iff $M,q \models_{\mathcal{L}} \varphi \wedge \psi$.
- $M,q \models_{\text{CSL}} tr(\langle\!\langle A\rangle\!\rangle_{\mathcal{K}(\Gamma)} \bigcirc \varphi)$ iff $M,q \models_{\text{CSL}} \hat{\mathcal{K}}_\Gamma \langle\!\langle A\rangle\!\rangle \bigcirc tr(\varphi)$ iff $M, \text{img}(q, \sim^{\mathcal{K}}_\Gamma) \models_{\text{CSL}} \langle\!\langle A\rangle\!\rangle \bigcirc tr(\varphi)$ iff $\exists_{S_A} \forall_{\lambda \in out(\text{img}(q,\sim^{\mathcal{K}}_\Gamma),S_A)} M, \lambda[1] \models_{\text{CSL}} tr(\varphi)$ iff (by induction) $\exists_{S_A} \forall_{\lambda \in out(\text{img}(q,\sim^{\mathcal{K}}_\Gamma),S_A)} M, \lambda[1] \models_{\text{ATOL}} \varphi$ iff $M,q \models_{\text{ATOL}} \langle\!\langle A\rangle\!\rangle_{\mathcal{K}(\Gamma)} \bigcirc \varphi$.
- For $\langle\!\langle A\rangle\!\rangle_{\mathcal{K}(\Gamma)} \Box\varphi$ and $\langle\!\langle A\rangle\!\rangle_{\mathcal{K}(\Gamma)} (\varphi \,\mathcal{U}\, \psi)$: analogously. The same for $\langle\!\langle A\rangle\!\rangle_{ir}\varphi$, $\langle\!\langle A\rangle\!\rangle^f_{\mathcal{K}}\varphi$, and $\langle\!\langle A\rangle\!\rangle^f_{K_b}\varphi$.
- $M,q \models_{\text{CSL}} tr(\langle\!\langle A\rangle\!\rangle^f \bigcirc \varphi)$ iff $M,q \models_{\text{CSL}} \langle\!\langle A\rangle\!\rangle \bigcirc tr(\varphi)$ iff $\exists_{S_A} \forall_{\lambda \in out(q,S_A)} M, \lambda[1] \models_{\text{CSL}} tr(\varphi)$ iff (by induction) $\exists_{S_A} \forall_{\lambda \in out(q,S_A)} M, \lambda[1] \models_{\text{F-ATEL}} \varphi$ iff $M,q \models_{\text{F-ATEL}} \langle\!\langle A\rangle\!\rangle^f \bigcirc \varphi$.
- $M,q \models_{\text{CSL}} tr(\langle\!\langle A\rangle\!\rangle^f_{M_b} \bigcirc \varphi)$ iff $M,q \models_{\text{CSL}} \neg K_b \neg \langle\!\langle A\rangle\!\rangle \bigcirc tr(\varphi)$ iff $\neg \forall_{q' \in \text{img}(q,\sim_b)} \neg M,q' \models_{\text{CSL}} \langle\!\langle A\rangle\!\rangle \bigcirc tr(\varphi)$ iff $\exists_{q' \in \text{img}(q,\sim_b)} \exists_{S_A} \forall_{\lambda \in out(q',S_A)} M, \lambda[1] \models_{\text{CSL}} tr(\varphi)$ iff (by induction) $\exists_{S_A} \exists_{q' \in \text{img}(q,\sim_b)} \forall_{\lambda \in out(q',S_A)} M, \lambda[1] \models_{\text{F-ATEL}} \varphi$ iff $M,q \models_{\text{F-ATEL}} \langle\!\langle A\rangle\!\rangle^f_{M_b} \bigcirc \varphi$.
- For $\langle\!\langle A\rangle\!\rangle^f_{M_b} \Box\varphi$ and $\langle\!\langle A\rangle\!\rangle^f_{M_b} (\varphi \,\mathcal{U}\, \psi)$: analogously.
- $M,q \models_{\text{CSL}} tr(\mathcal{K}_A\varphi)$ iff $M,q \models_{\text{CSL}} \mathcal{K}_A tr(\varphi)$ iff $\forall_{q' \in \text{img}(q,\sim^{\mathcal{K}}_A)} M,q' \models_{\text{CSL}} tr(\varphi)$ iff (by induction) $\forall_{q' \in \text{img}(q,\sim^{\mathcal{K}}_A)} M,q' \models_{\mathcal{L}} \varphi$ iff $M,q \models_{\mathcal{L}} \mathcal{K}_A\varphi$.

■

Theorem 4

Proof (structural induction on the structure of φ) In each case, we will prove that $M, Q \models \mathbb{K}_a\varphi$ implies $M, Q \models \varphi$ for an arbitrary Q. By Proposition 3, we can then conclude that $M, Q \models \mathbb{K}_a\varphi \rightarrow \varphi$.

To simplify the proof, we assume that each φ has been transformed so that no constructive knowledge operator is followed by conjunction (by Proposition 15.3, each subformula $\hat{\mathcal{K}}_A(\psi_1 \wedge \psi_2)$ can be equivalently transformed to $\hat{\mathcal{K}}_A\varphi_1 \wedge \hat{\mathcal{K}}_A\varphi_2$, and we can apply this transformation recursively). Thus, every $\hat{\mathcal{K}}$ in φ is now followed either by some other $\hat{\mathcal{K}}'$, or by $\langle\langle A\rangle\rangle$, or by a standard knowledge operator $\mathcal{K}''$, or by an atomic proposition p.

Additionally, given Q, we define $Q' = \mathrm{img}(Q, \sim_a)$. Note that $Q \subseteq Q'$ by reflexivity of $\sim_a$. Also, $out(Q, S_A) \subseteq out(Q', S_A)$ by monotonicity of function out wrt Q, and $\mathrm{img}(Q, \sim_A^{\mathcal{K}}) \subseteq \mathrm{img}(Q', \sim_A^{\mathcal{K}})$ by reflexivity of all $\sim_A^{\mathcal{K}}$.

Case $\varphi \equiv p$**:** Let $M, Q \models \mathbb{K}_a p$. Then $M, Q' \models p$, i.e. $\forall_{q \in Q'} M, q \models p$. So, $\forall_{q \in Q} M, q \models p$, and $M, Q \models p$.

Case $\varphi \equiv \psi_1 \wedge \psi_2$**:** Let $M, Q \models \mathbb{K}_a(\psi_1 \wedge \psi_2)$. Then $M, Q' \models \psi_1 \wedge \psi_2$, i.e. $M, Q' \models \psi_1$ and $M, Q' \models \psi_2$. So, $M, Q \models \mathbb{K}_a\psi_1$ and $M, Q \models \mathbb{K}_a\psi_2$. By the induction hypothesis, $M, Q \models \psi_1$ and $M, Q \models \psi_2$, and hence $M, Q \models \psi_1 \wedge \psi_2$.

Case $\varphi \equiv \langle\langle A\rangle\rangle\bigcirc\psi$**:** Let $M, Q \models \mathbb{K}_a\langle\langle A\rangle\rangle\bigcirc\psi$. Then $M, Q' \models \langle\langle A\rangle\rangle\bigcirc\psi$, and so $\exists_{S_A}\forall_{\lambda \in out(Q', S_A)} M, \lambda[1] \models \psi$. Thus, $\exists_{S_A}\forall_{\lambda \in out(Q, S_A)} M, \lambda[1] \models \psi$, and $M, Q \models \langle\langle A\rangle\rangle\bigcirc\psi$.

Cases $\varphi \equiv \langle\langle A\rangle\rangle\Box\psi$ **and** $\varphi \equiv \langle\langle A\rangle\rangle\psi_1\,\mathcal{U}\,\psi_2$**:** analogous.

Case $\varphi \equiv \mathcal{K}_A\psi$**:** Let $M, Q \models \mathbb{K}_a\mathcal{K}_A\psi$. Then $M, Q' \models \mathcal{K}_A\psi$, and $\forall_{q \in \mathrm{img}(Q', \sim_A^{\mathcal{K}})} M, q \models \psi$. But then also $\forall_{q \in \mathrm{img}(Q, \sim_A^{\mathcal{K}})} M, q \models \psi$, and $M, Q \models \mathcal{K}_A\psi$.

Before we consider the remaining cases, we define a couple of additional symbols. Let $Q^i = \mathrm{img}(Q^{i-1}, \sim_{A_i}^{\mathcal{K}^i})$, with the initial set $Q^0 = Q$. That is, $Q^i = \mathrm{img}(...(\mathrm{img}(Q, \sim_{A_1}^{\mathcal{K}^1}), ...), \sim_{A_i}^{\mathcal{K}^i})$. Also, let $Q'' = \mathrm{img}(Q^n, \sim_a)$. Note that $Q^n \subseteq Q''$, and $out(Q^n, S_B) \subseteq out(Q'', S_B)$ for any S_B.

Case $\varphi \equiv \hat{\mathcal{K}}^n_{A_n}...\hat{\mathcal{K}}^1_{A_1}p$**:** (i.e., φ is a sequence of n possibly different $\hat{\mathcal{K}}$ operators for possibly different coalitions). Let $M, Q \models \mathbb{K}_a\hat{\mathcal{K}}^n_{A_n}...\hat{\mathcal{K}}^1_{A_1}p$. Then $M, Q'' \models p$, and hence $\forall_{q \in Q''} M, q \models p$. Thus, $\forall_{q \in Q^n} M, q \models p$, so $M, Q^n \models p$, and $M, Q \models \hat{\mathcal{K}}^n_{A_n}...\hat{\mathcal{K}}^1_{A_1}p$.

Case $\varphi \equiv \hat{\mathcal{K}}^n_{A_n}...\hat{\mathcal{K}}^1_{A_1}\mathcal{K}_B\psi$**:** analogous.

Case $\varphi \equiv \hat{\mathcal{K}}^n_{A_n}...\hat{\mathcal{K}}^1_{A_1}\langle\langle B\rangle\rangle\bigcirc\psi$**:** Let $M, Q \models \mathbb{K}_a\hat{\mathcal{K}}^n_{A_n}...\hat{\mathcal{K}}^1_{A_1}\langle\langle B\rangle\rangle\bigcirc\psi$.
Then $M, Q'' \models \langle\langle B\rangle\rangle\bigcirc\psi$, and hence $\exists_{S_B}\forall_{\lambda \in out(Q'', S_B)} M, \lambda[1] \models \psi$.
Thus, $\exists_{S_B}\forall_{\lambda \in out(Q^n, S_B)} M, \lambda[1] \models \psi$, so $M, Q^n \models \langle\langle B\rangle\rangle\bigcirc\psi$, and $M, Q \models \hat{\mathcal{K}}^n_{A_n}...\hat{\mathcal{K}}^1_{A_1}\langle\langle B\rangle\rangle\bigcirc\psi$.

Cases $\varphi \equiv \hat{\mathcal{K}}^n_{A_n}...\hat{\mathcal{K}}^1_{A_1}\langle\langle B\rangle\rangle\Box\psi$ **and** $\varphi \equiv \hat{\mathcal{K}}^n_{A_n}...\hat{\mathcal{K}}^1_{A_1}\langle\langle B\rangle\rangle\psi_1\,\mathcal{U}\,\psi_2$**:** analogous.

■

Proposition 16

Proof

K Immediate.

D Suppose that $M, Q \models \mathsf{loc}\,\bot$ for some $Q \neq \emptyset$. Then, there is some q for which $M, q \models \bot$, but this contradicts Proposition 3.1.

T To see that **T** is not strongly valid, let φ and M be as in Lemma 1.1, and take $Q = \mathrm{img}(q, \sim_a)$. $M, Q \models \mathsf{loc}\,\varphi$, but $M, Q \not\models \varphi$.

4/**4**$^+$ $M, Q \models \mathsf{loc}\,\varphi$ iff for every $q \in Q$ we have that $M, q \models \varphi$ iff for every $q \in Q$ we have that $M, q \models \mathsf{loc}\,\varphi$ iff $M, Q \models \mathsf{loc}\,\mathsf{loc}\,\varphi$.

5/**5**$^+$ To see that **5** is not strongly valid, let M be as in Figure 2.6.2, $\varphi = p$ and let $Q = \{q, q'\}$. $M, Q \models \neg\mathsf{loc}\,\varphi$ because $M, q' \models \neg\varphi$. However, if it were the case that $M, Q \models \mathsf{loc}\,\neg\mathsf{loc}\,\varphi$, then $M, q \models \neg\mathsf{loc}\,\varphi$ and thus, $M, q \models \neg\varphi$, which is not the case.

B $M, Q \models \varphi$ iff for all $q \in Q$ we have that $M, q \models \varphi$ iff for all $q \in Q$ we have that $M, q \models \neg\mathsf{loc}\,\neg\varphi$ iff $M, Q \models \mathsf{loc}\,\neg\mathsf{loc}\,\neg\varphi$.

■

Theorem 7

Proof

∂**K:** We construct a counterexample. Let M be a model with states q_1, q_2 and agent a, such that $q_1 \sim_a q_2$, $\pi(q_1) = \{r\}$ and $\pi(q_2) = \{p\}$. Let $\varphi = \neg p$ and $\psi = r$. $p \notin \pi(q_1) \cap \pi(q_2)$, so $M, \mathrm{img}(q_1, \sim_a) \models \varphi$ and $M, q_1 \models \mathbb{K}_a\varphi$. $r \notin \pi(q_1) \cap \pi(q_2)$, so $M, \mathrm{img}(q_1, \sim_a) \not\models \psi$ and $M, q_1 \not\models \mathbb{K}_a\psi$. Thus, $M, q_1 \not\models \mathbb{K}_a\varphi \rightarrow \mathbb{K}_a\psi$ and by Proposition 25: $M, q_1 \not\models \mathbb{K}_a\varphi \rightsquigarrow \mathbb{K}_a\psi$ (*). Since both $M, q_1 \models \varphi \rightarrow \psi$ and $M, q_2 \models \varphi \rightarrow \psi$, by Proposition 22, $M, \mathrm{img}(q_1, \sim_a) \models \varphi \rightsquigarrow \psi$ and thus $M, q_1 \models \mathbb{K}_a(\varphi \rightsquigarrow \psi)$. Together with (*), we get that $M, q_1 \not\models \mathbb{K}_a(\varphi \rightsquigarrow \psi) \rightarrow (\mathbb{K}_a\varphi \rightsquigarrow \mathbb{K}_a\psi)$ and, by Proposition 25, $M, q_1 \not\models \mathbb{K}_a(\varphi \rightsquigarrow \psi) \rightsquigarrow (\mathbb{K}_a\varphi \rightsquigarrow \mathbb{K}_a\psi)$. Thus, ∂**K** is not weakly (and hence not strongly) valid.

∂**T:** Let M, q, a, φ be as in Lemma 1.2. $M, q \models \mathbb{K}_a\varphi$ and $M, q \not\models \varphi$, so $M, q \not\models \mathbb{K}_a\varphi \rightsquigarrow \varphi$ by Proposition 25. Thus, ∂**T** is not weakly (and hence not strongly) valid.

∂**4**$^+$/∂**4:** $M, Q \models \mathbb{K}_a\varphi \leftrightsquigarrow \mathbb{K}_a\mathbb{K}_a\varphi$ iff, by Proposition 22, $\forall_{q\in Q}(M, q \models \mathbb{K}_a\varphi \Leftrightarrow M, q \models \mathbb{K}_a\mathbb{K}_a\varphi)$ iff, by **4**$^+$, $\forall_{q\in Q}(M, q \models \mathbb{K}_a\varphi \Leftrightarrow M, q \models \mathbb{K}_a\varphi)$.

∂**5**$^+$/∂**5:** $M, Q \models \sim\mathbb{K}_a\varphi \leftrightsquigarrow \mathbb{K}_a \sim\mathbb{K}_a\varphi$ iff, by Proposition 22, $\forall_{q\in Q}(M, q \models \sim\mathbb{K}_a\varphi \Leftrightarrow M, q \models \mathbb{K}_a \sim\mathbb{K}_a\varphi)$ iff $\forall_{q\in Q}(M, \mathrm{img}(q, \sim_a) \not\models \varphi \Leftrightarrow M, \mathrm{img}(q, \sim_a) \models \sim\mathbb{K}_a\varphi)$ iff $\forall_{q\in Q}(M, \mathrm{img}(q, \sim_a) \not\models \varphi \Leftrightarrow \forall_{q'\in \mathrm{img}(q,\sim_a)} M, \mathrm{img}(q', \sim_a) \not\models \varphi)$ which is true, since $\mathrm{img}(q', \sim_a) = \mathrm{img}(q, \sim_a)$ for any $q' \in \mathrm{img}(q, \sim_a)$.

∂**D:** $M, Q \models \sim\mathbb{K}_a\lightning$ iff $\forall_{q\in Q} M, q \not\models \mathbb{K}_a\lightning$ iff $\forall_{q\in Q} M, \mathrm{img}(q, \sim_a) \not\models \lightning$, which is true by Proposition 23.1.

∂**B:** $M, Q \models \varphi \rightsquigarrow \mathbb{K}_a \sim \mathbb{K}_a \sim \varphi$ iff $\forall_{q \in Q}(M, q \models \varphi \Rightarrow M, \mathrm{img}(q, \sim_a) \models \sim \mathbb{K}_a \sim \varphi)$ iff $\forall_{q \in Q}(M, q \models \varphi \Rightarrow \forall_{q' \in \mathrm{img}(q, \sim_a)} M, q' \not\models \mathbb{K}_a \sim \varphi)$ iff $\forall_{q \in Q}(M, q \models \varphi \Rightarrow \forall_{q' \in \mathrm{img}(q, \sim_a)} M, \mathrm{img}(q', \sim_a) \not\models \sim \varphi)$ iff $\forall_{q \in Q}(M, q \models \varphi \Rightarrow \forall_{q' \in \mathrm{img}(q, \sim_a)} \exists_{q'' \in \mathrm{img}(q', \sim_a)} M, q'' \models \varphi)$ iff $\forall_{q \in Q}(M, q \models \varphi \Rightarrow \exists_{q' \in \mathrm{img}(q, \sim_a)} M, q' \models \varphi)$. This always holds, by taking $q' = q$.

■

Theorem 9 and Corollary 4 (Constructive Normal Form) and Theorem 10 (Expressiveness of Strong Negation)

In the following we will very often work in the language $\mathcal{L}(\neg, \wedge, \langle\langle A \rangle\rangle T, \hat{\mathcal{K}}_A, \sim)$, and we will henceforth use the shorthand notation $\hat{\mathcal{L}}$ to denote this language, for simplicity.

We use $Subf(\varphi)$ to denote the set of all subformulae of φ (including φ itself). For simplicity, we assume that each subformula of a formula is *unique*, i.e. that there is a unique member of $Subf(\varphi)$ for each occurrence of a subformula in φ[13].

We first present intermediate definitions and results leading up to the main result in Theorem 9. Note that Lemma 3 below gives an alternative (equivalent) definition of constructive normal form.

Definition 2 *We define the* depth $d_\varphi(\psi)$ *of a subformula* $\psi \in Subf(\varphi)$ *of a formula* $\varphi \in \hat{\mathcal{L}}$ *in the usual way:*

- $d_\varphi(\varphi) = 0$
- $d_\varphi(\hat{\mathcal{K}}\gamma) = d \Rightarrow d_\varphi(\gamma) = d + 1$
- $d_\varphi(\langle\langle G \rangle\rangle T\gamma) = d \Rightarrow d_\varphi(\gamma) = d + 1$
- $d_\varphi(\gamma_1 \wedge \gamma_2) = d \Rightarrow d_\varphi(\gamma_1) = d_\varphi(\gamma_2) = d + 1$
- $d_\varphi(\neg\gamma) = d \Rightarrow d_\varphi(\gamma) = d + 1$
- $d_\varphi(\sim\gamma) = d \Rightarrow d_\varphi(\gamma) = d + 1$

Lemma 2 *A formula* $\psi \in \hat{\mathcal{L}}$ *is of CSNF iff every* $\gamma \in Subf(\psi)$ *is of CSNF.*

Proof The implication to the left is trivial; we prove the one to the right. Assume that ψ is of CSNF. That each $\gamma \in Subf(\psi)$ is of CSNF follows immediately by induction on the depth of γ:

- $d_\psi(\gamma) = 0$: $\gamma = \psi$ is of CSNF
- $d_\psi(\gamma) = d + 1$ ($d \geq 0$). We reason by the possible cases:
  - $d_\psi(\hat{\mathcal{K}}\gamma) = d$: by the induction hypothesis $\hat{\mathcal{K}}\gamma$ is of CSNF, and thus γ is of CSNF.

[13] This can be achieved by, e.g., adorning the subformulae with unique identifiers, or by taking $Subf(\varphi)$ to be a multiset instead of a set. The only reason for this assumption is to make proofs simpler.

- $d_\psi(\langle\langle G\rangle\rangle T\gamma) = d$: $\langle\langle G\rangle\rangle T\gamma$ is of CSNF; γ is of CSNF.
- $d_\psi(\gamma \wedge \gamma') = d$: $\gamma \wedge \gamma'$ is of CSNF; γ is of CSNF. Similarly when $d_\psi(\gamma' \wedge \gamma) = d$.
- $d_\psi(\neg\gamma) = d$: $\neg\gamma$ is of CSNF; γ is of CSNF.
- $d_\psi(\sim\gamma) = d$: $\sim\gamma$ is of CSNF; γ is of CSNF.

■

Lemma 3 *A formula $\psi \in \hat{\mathcal{L}}$ is of CSNF iff every $\hat{\mathcal{K}}\gamma \in Subf(\psi)$ is of the form $\hat{\mathcal{K}}\hat{\mathcal{K}}_0 \cdots \hat{\mathcal{K}}_k\alpha$ where $\alpha \in Atoms$, for some $k \geq 0$.*

Proof For the direction to the right, assume that there is a $\hat{\mathcal{K}}\gamma \in Subf(\psi)$ which is *not* of the form. There are two possibilities: $\gamma = \hat{\mathcal{K}}_0 \cdots \hat{\mathcal{K}}_m\neg\beta$ or $\gamma = \hat{\mathcal{K}}_0 \cdots \hat{\mathcal{K}}_m\beta_1 \wedge \beta_2$ for some $m \geq 0$. In either case, it follows immediately that $\hat{\mathcal{K}}\gamma$ is not of CSNF. By Lemma 2, ψ is not of CSNF.

For the direction to the left, assume that every $\hat{\mathcal{K}}\gamma \in Subf(\psi)$ is of the form. We show that every $\chi \in Subf(\psi)$ is of CSNF by structural induction:

- $\chi = p \in \Theta$: χ is of CSNF.
- $\chi = \hat{\mathcal{K}}\gamma$: by the induction hypothesis, γ is of CSNF. By assumption, $\gamma = \hat{\mathcal{K}}_0 \cdots \hat{\mathcal{K}}_k\alpha$ for some $\alpha \in Atoms$ and some $k \geq 0$. Thus, χ is of CSNF.
- $\chi = \langle\langle A\rangle\rangle T\gamma$: by the induction hypothesis, γ is of CSNF, and thus χ is of CSNF.
- $\chi = \gamma_1 \wedge \gamma_2$: by the induction hypothesis, γ_1 and γ_2 are of CSNF, and thus χ is of CSNF.
- $\chi = \sim\gamma$: by the induction hypothesis, γ is of CSNF and thus χ is of CSNF.
- $\chi = \neg\gamma$: by the induction hypothesis, γ is of CSNF and thus χ is of CSNF.

■

Now that we have established some properties of formulae of CSNF, we go on to define the mapping of a formula to one of CSNF.

Definition 3 *The value $f(\hat{\mathcal{K}}\psi)$ of the function $f : \{\hat{\mathcal{K}}\psi : \psi \in \hat{\mathcal{L}}\} \to \hat{\mathcal{L}}$ is defined by structural induction over ψ:*

$$
\begin{aligned}
f(\hat{\mathcal{K}}\psi) &= \hat{\mathcal{K}}\psi \quad \textit{when } \psi \in Atoms \\
f(\hat{\mathcal{K}}\hat{\mathcal{K}}'\gamma) &= \hat{\mathcal{K}}\hat{\mathcal{K}}'\gamma \\
f(\hat{\mathcal{K}}\neg\gamma) &= \neg f(\hat{\mathcal{K}}\gamma) \\
f(\hat{\mathcal{K}}(\gamma_1 \wedge \gamma_2)) &= f(\hat{\mathcal{K}}\gamma_1) \wedge f(\hat{\mathcal{K}}\gamma_2)
\end{aligned}
$$

Lemma 4 *Let $\beta \in \hat{\mathcal{L}}$ be a formula. β is of CSNF iff $f(\hat{\mathcal{K}}\beta)$ is of CSNF for any arbitrary $\hat{\mathcal{K}} \in \{\mathbb{C}_A, \mathbb{D}_A, \mathbb{E}_A : A \subseteq \Sigma\}$.*

Proof Let $\hat{\mathcal{K}} \in \{\mathbb{C}_A, \mathbb{D}_A, \mathbb{E}_A : A \subseteq \Sigma\}$. The proof is by structural induction over β:

- $\beta = p \in \Theta$: β is of CSNF iff $f(\hat{\mathcal{K}}\beta) = \hat{\mathcal{K}}p$ is of CSNF.
- $\beta = \hat{\mathcal{K}}'\gamma$: β is of CSNF iff $f(\hat{\mathcal{K}}\beta) = \hat{\mathcal{K}}\hat{\mathcal{K}}'\gamma$ is of CSNF.
- $\beta = \langle\langle A\rangle\rangle T\gamma$: β is of CSNF iff $f(\hat{\mathcal{K}}\beta) = \hat{\mathcal{K}}\langle\langle A\rangle\rangle T\gamma$ is of CSNF.
- $\beta = \gamma_1 \wedge \gamma_2$: β is of CSNF iff both γ_1 and γ_2 are of CSNF iff, by the induction hypothesis, both $f(\hat{\mathcal{K}}\gamma_1)$ and $f(\hat{\mathcal{K}}\gamma_2)$ are of CSNF iff $f(\hat{\mathcal{K}}\beta)$ is of CSNF.
- $\beta = {\sim}\gamma$: β is of CSNF iff $f(\hat{\mathcal{K}}\beta) = \hat{\mathcal{K}}\beta$ is of CSNF.
- $\beta = \neg\gamma$: β is of CSNF iff γ is of CSNF iff, by the induction hypothesis, $f(\hat{\mathcal{K}}\gamma)$ is of CSNF iff $\neg f(\hat{\mathcal{K}}\gamma)$ is of CSNF iff $f(\hat{\mathcal{K}}\neg\gamma)$ is of CSNF.

■

Lemma 5 *For any $\psi \in \hat{\mathcal{L}}$,*

$$\hat{\mathcal{K}}\psi \leftrightarrow f(\hat{\mathcal{K}}\psi)$$

is strongly valid for any $\hat{\mathcal{K}} \in \{\mathbb{C}_A, \mathbb{D}_A, \mathbb{E}_A : A \subseteq \Sigma\}$.

Proof The proof is by structural induction over ψ. When $\psi \in Atoms$ or $\psi = \hat{\mathcal{K}}'\gamma$, $f(\hat{\mathcal{K}}\psi) = \hat{\mathcal{K}}\psi$, and we are done. When $\psi = \neg\gamma$, $M, Q \models \hat{\mathcal{K}}\psi$ iff, by Proposition 15, $M, Q \models \neg\hat{\mathcal{K}}\gamma$ iff $M, Q \not\models \hat{\mathcal{K}}\gamma$ iff, by the induction hypothesis, $M, Q \not\models f(\hat{\mathcal{K}}\gamma)$ iff $M, Q \models \neg f(\hat{\mathcal{K}}\gamma)$ iff $M, Q \models f(\hat{\mathcal{K}}\psi)$. When $\psi = \gamma_1 \wedge \gamma_2$, $M, Q \models \hat{\mathcal{K}}\psi$ iff, by Proposition 15, $M, Q \models \hat{\mathcal{K}}\gamma_1$ and $M, Q \models \hat{\mathcal{K}}\gamma_2$ iff, by the induction hypothesis, $M, Q \models f(\hat{\mathcal{K}}\gamma_1)$ and $M, Q \models f(\hat{\mathcal{K}}\gamma_2)$ iff $M, Q \models f(\hat{\mathcal{K}}\psi)$. ■

Definition 4 (φ_i, X_i, α_i) *Let $\varphi \in \hat{\mathcal{L}}$ be a formula. Define φ_i, $i \geq 0$:*

- *$i = 0$: $\varphi_0 = \varphi$*
- *$i = j+1$ ($j \geq 0$): Let $X_i = \{\hat{\mathcal{K}}\psi : \hat{\mathcal{K}}\psi \in Subf(\varphi_j), \hat{\mathcal{K}}\psi$ is not of CSNF$\}$. If X_i is empty, let $\varphi_{j+1} = \varphi_j$. Otherwise, select an $\alpha_i \in X_i$ such that $\beta \in X_i$ implies that $d_{\varphi_j}(\beta) \leq d_{\varphi_j}(\alpha_i)$ (several such α_i may exist; select one arbitrarily), and let φ_{j+1} be φ_j with the subformula α_i replaced by $f(\alpha_i)$.*

Lemma 6 *Let $\varphi \in \hat{\mathcal{L}}$, and let α_i be defined in Def. 4. For each $i \geq 1$, $f(\alpha_i)$ is of CSNF.*

Proof Let $\alpha_i = \hat{\mathcal{K}}\psi$. We show that for every $\gamma \in Subf(\psi)$, $f(\hat{\mathcal{K}}\gamma)$ is of CSNF by structural induction over γ:

- $\gamma = p \in \Theta$: $f(\hat{\mathcal{K}}\gamma) = \hat{\mathcal{K}}p$ is of CSNF.
- $\gamma = \hat{\mathcal{K}}'\beta$: $f(\hat{\mathcal{K}}\gamma) = \hat{\mathcal{K}}\hat{\mathcal{K}}'\beta$ is of CSNF iff $\hat{\mathcal{K}}'\beta$ is of CSNF. Assume that $\hat{\mathcal{K}}'\beta$ is *not* of CSNF, then $\gamma \in X_i$. Then $d_{\varphi_{i-1}}(\gamma) > d_{\varphi_{i-1}}(\alpha_i)$, but this is a contradiction since there are no $\gamma \in X_i$ with greater depth than α_i. Thus, $f(\hat{\mathcal{K}}\gamma)$ is of CSNF.

- $\gamma = \langle\langle G\rangle\rangle T\beta$: By the induction hypothesis, $f(\hat{\mathcal{K}}\beta)$ is of CSNF; by Lemma 4 β is of CSNF; $\langle\langle G\rangle\rangle T\beta$ is of CSNF; $\hat{\mathcal{K}}\langle\langle G\rangle\rangle T\beta = f(\hat{\mathcal{K}}\gamma)$ is of CSNF.
- $\gamma = \gamma_1 \wedge \gamma_2$: By the induction hypothesis, $f(\hat{\mathcal{K}}\gamma_1)$ and $f(\hat{\mathcal{K}}\gamma_2)$ are of CSNF; $f(\hat{\mathcal{K}}\gamma_1) \wedge f(\hat{\mathcal{K}}\gamma_2)$ is of CSNF; $f(\hat{\mathcal{K}}\gamma)$ is of CSNF.
- $\gamma = \neg\beta$: By the induction hypothesis, $f(\hat{\mathcal{K}}\beta)$ is of CSNF; $\neg f(\hat{\mathcal{K}}\beta)$ is of CSNF; $f(\hat{\mathcal{K}}\gamma)$ is of CSNF.
- $\gamma = \sim\beta$: By the induction hypothesis, $f(\hat{\mathcal{K}}\beta)$ is of CSNF; by Lemma 4 β is of CSNF; γ is of CSNF; $\hat{\mathcal{K}}\gamma = f(\hat{\mathcal{K}}\gamma)$ is of CSNF.

■

Lemma 7 *Let $\varphi \in \hat{\mathcal{L}}$, and let φ_i be defined in Def. 4. There is a $p \geq 1$ such that $\varphi_p = \varphi_{p-1}$ and $X_p = \emptyset$. We write $\hat{\varphi} = \varphi_p$ for an arbitrary such p.*

Proof X_1 is finite. We show that $X_{i+1} \subset X_i$ (proper inclusion) whenever $\varphi_i \neq \varphi_{i-1}$, for any $i \geq 1$. The Lemma follows.

Let $\varphi_i \neq \varphi_{i-1}$. Assume that there is an $\alpha \in X_{i+1}$, $\alpha \notin X_i$. α is not of CSNF, and since $\alpha \in \mathit{Subf}(\varphi_i)$ and $\alpha \notin \mathit{Subf}(\varphi_{i-1})$ the only possibility is that $\alpha \in \mathit{Subf}(f(\alpha_i))$. But by Lemma 6, $f(\alpha_i)$ is of CSNF, and by Lemma 2 α must be of CSNF which is a contradiction. Thus, $X_{i+1} \subseteq X_i$. To see that the inclusion is proper, observe that $\alpha_i \in X_i$ but $\alpha_i \notin X_{i+1}$. ■

Proof of Theorem 9 Let $\varphi'' \in \mathcal{L}^*$, and let φ' be the result of replacing every occurrence of $\mathcal{K}$ in φ'' with the combination $\hat{\mathcal{K}} \sim\!\sim$, for every $\mathcal{K}$. Let φ be the result of replacing every occurrence of loc in φ' with the combination $\sim\!\sim$. φ'' and φ are strongly equivalent by Remark 7. Observe that $\varphi \in \hat{\mathcal{L}}$. Let $\hat{\varphi} = \varphi_p$ be defined from φ as in Lemma 7.

First, we argue that $\hat{\varphi}$ is of CSNF. If not, there is a $\hat{\mathcal{K}}\gamma \in \mathit{Subf}(\hat{\varphi})$ where γ is not of the form $\hat{\mathcal{K}}_0 \cdots \hat{\mathcal{K}}_k\alpha$ for $\alpha \in \mathit{Atoms}$ (Lemma 3). Then, $\hat{\mathcal{K}}\gamma$ is not of CSNF, which contradicts the fact that $X_p = \emptyset$. Second, we show that $\hat{\varphi} \leftrightarrow \varphi$ is strongly valid. Let $i \geq 1$. By Lemma 5, $M, Q \models \alpha_i$ iff $M, Q \models f(\alpha_i)$ for any M, Q. It follows immediately that $M, Q \models \varphi_i$ iff $M, Q \models \varphi_{i+1}$. Thus, $M, Q \models \varphi = \varphi_0$ iff $M, Q \models \hat{\varphi} = \varphi_p$. Thus, $\hat{\varphi}$ is of CSNF, and it is equivalent to φ. ■

Proof of Corollary 4 Let $\varphi \in \mathcal{L}^*$. By the theorem, φ is strongly equivalent to a formula $\hat{\varphi}$ which is of CSNF. Now, we recursively replace all subformulae of $\hat{\varphi}$ of the form $\sim\psi$ with $\langle\langle\emptyset\rangle\rangle(\neg\psi)\,\mathcal{U}\,(\neg\psi)$, yielding (by Proposition 18) a strongly equivalent formula φ' without strong negation. We observe that subformulae of CSNF are replaced with subformulae of CSNF, so φ' is of CSNF too. ■

We now go on to present our proof of Theorem 10. Some more notation: when $\varphi \in \hat{\mathcal{L}}$, we use $\partial\varphi$ to denote the result of replacing each occurrence of $\neg$ in φ with $\sim$. Formally, $\partial p = p$; $\partial\mathbb{K}_a\psi = \mathbb{K}_a\partial\psi$; $\partial\langle\langle G\rangle\rangle T\psi = \langle\langle G\rangle\rangle T\partial\psi$; $\partial\psi_1 \wedge \psi_2 = \partial\psi_1 \wedge \partial\psi_2$; $\partial\neg\psi = \sim\partial\psi$; $\partial\sim\psi = \sim\partial\psi$.

We begin with defining the notion of *constructive depth* of a subformula – not to be confused with the notion of *depth* in the proof of Theorem 9.

Definition 5 (Constructive depth) *Let $\varphi \in \hat{\mathcal{L}}$. The* constructive depth, *or just c-depth, $D_\varphi(\psi)$ in φ of a subformula $\psi \in Subf(\varphi)$ is defined inductively as follows:*

- $D_\varphi(\varphi) = 0$
- $D_\varphi(\hat{\mathcal{K}}\gamma) = D \Rightarrow D_\varphi(\gamma) = D + 1$
- $D_\varphi(\langle\!\langle G\rangle\!\rangle T\gamma) = D \Rightarrow D_\varphi(\gamma) = 0$
- $D_\varphi(\gamma_1 \wedge \gamma_2) = D \Rightarrow D_\varphi(\gamma_1) = D_\varphi(\gamma_2) = D$
- $D_\varphi(\neg\gamma) = D \Rightarrow D_\varphi(\gamma) = D$
- $D_\varphi(\sim\gamma) = D \Rightarrow D_\varphi(\gamma) = 0$

If φ has no occurrence of $\neg$ on c-depth D, i.e., if $\neg\psi \in Subf(\varphi)$ implies that $D_\varphi(\neg\psi) \neq D$, we say that φ is *free of $\neg$ on depth D*.

Lemma 8 *If a formula $\varphi \in \mathcal{L}(\neg, \wedge, \langle\!\langle A\rangle\!\rangle T, \hat{\mathcal{K}}, \sim)$ is free of $\neg$ on all depths > 0, then*

$$\varphi \leftrightarrow \partial\varphi$$

is valid.

Proof We show that

$$\psi \leftrightarrow \partial\psi \text{ is } \begin{cases} \text{valid} & \text{if } D_\varphi(\psi) = 0 \\ \text{strongly valid} & \text{if } D_\varphi(\psi) > 0 \end{cases}$$

for all $\psi \in Subf(\varphi)$ by structural induction.

$\psi = p$**:** immediate ($\partial\psi = \psi$).

$\psi = \hat{\mathcal{K}}_A\gamma$**:** $D_\varphi(\gamma) > 0$. $M, Q \models \psi$ iff $M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}) \models \gamma$ iff, by the induction hypothesis, $M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}) \models \partial\gamma$ iff $M, Q \models \mathcal{K}_A\partial\gamma$ iff $M, Q \models \partial\psi$.

$\psi = \langle\!\langle G\rangle\!\rangle\Box\gamma$**:** $M, Q \models \psi$ iff $\exists_{S_G}\forall_{\lambda\in out(Q,S_G)}\forall_{j\geq 0} M, \lambda[j] \models \gamma$ iff, by the induction hypothesis (for γ, where $D_\varphi(\gamma) = 0$), $\exists_{S_G}\forall_{\lambda\in out(Q,S_G)}\forall_{j\geq 0} M, \lambda[j] \models \partial\gamma$ iff $M, Q \models \langle\!\langle G\rangle\!\rangle\Box\partial\gamma$. Similar for the other ATL connectives.

$\psi = \gamma_1 \wedge \gamma_2$**:** First, consider the case that $D_\varphi(\psi) = 0$, in which case $D_\varphi(\gamma_1) = D_\varphi(\gamma_2) = 0$. We must show that $\psi \leftrightarrow \partial\psi$ is valid. $M, q \models \psi$ iff $M, q \models \gamma_1$ and $M, q \models \gamma_2$ iff, by the induction hypothesis, $M, q \models \partial\gamma_1$ and $M, q \models \partial\gamma_2$ iff $M, q \models \partial\gamma_1 \wedge \partial\gamma_2$. Second, consider the case that $D_\varphi(\psi) > 0$, in which case $D_\varphi(\gamma_1) > 0$ and $D_\varphi(\gamma_2) > 0$. We must show that $\psi \leftrightarrow \partial\psi$ is strongly valid. $M, Q \models \psi$ iff $M, Q \models \gamma_1$ and $M, Q \models \gamma_2$ iff, by the induction hypothesis, $M, Q \models \partial\gamma_1$ and $M, Q \models \partial\gamma_2$ iff $M, Q \models \partial\gamma_1 \wedge \partial\gamma_2$.

$\psi = \neg\gamma$**:** By the assumption in the lemma, $D_\varphi(\psi) = 0$. Then also $D_\varphi(\gamma) = 0$. $M, q \models \psi$ iff $M, q \not\models \gamma$ iff, by the induction hypothesis, $M, q \not\models \partial\gamma$ iff $M, q \models \sim\partial\gamma$.

$\psi = \sim\gamma$**:** $M, Q \models \psi$ iff $\forall_{q\in Q} M, q \not\models \gamma$ iff, by the induction hypothesis (for γ, where $D_\varphi(\gamma) = 0$), $\forall_{q\in Q} M, q \not\models \partial\gamma$ iff $M, Q \models \sim\partial\gamma$.

■

Proof of Theorem 10 Let $\varphi \in \mathcal{L}(\neg, \wedge, \langle\langle A\rangle\rangle T, \hat{\mathcal{K}}, \sim)$ be a formula, and let $\hat{\varphi}$ be a formula of CSNF equivalent to φ. Note that $\hat{\varphi} \in \mathcal{L}(\neg, \wedge, \langle\langle A\rangle\rangle T, \hat{\mathcal{K}}, \sim)$. We show that

$$D_{\hat{\varphi}}(\psi) > 0 \Rightarrow \psi \in \mathit{Atoms} \text{ or } \psi = \hat{\mathcal{K}}\gamma \tag{2.3}$$

for every $\psi \in \mathit{Subf}(\hat{\varphi})$ by induction over the depth (not the constructive depth) of ψ, for arbitrary γ and $\hat{\mathcal{K}}$. For the base case, let $\psi = \hat{\varphi}$ and (2.3) is vacuously true. In the inductive case assume that (2.3) holds for the parent of ψ. There are three circumstances in which $D_{\hat{\varphi}}(\psi) > 0$. First, $\hat{\mathcal{K}}\psi \in \mathit{Subf}(\hat{\varphi})$. Then, $\psi \in \mathit{Atoms}$ or ψ is of the form $\mathbb{K}_b\gamma$, since $\hat{\varphi}$ is of CSNF. Second, $\neg\psi \in \mathit{Subf}(\hat{\varphi})$ with $D_{\hat{\varphi}}(\psi) = D_{\hat{\varphi}}(\neg\psi)$. By the induction hypothesis, it must be the case that $D_{\hat{\varphi}}(\neg\psi) = 0$, so (2.3) is vacuously true. Third, $\psi \wedge \psi' \in \mathit{Subf}(\hat{\varphi})$ with $D_{\hat{\varphi}}(\psi) = D_{\hat{\varphi}}(\psi') = D_{\hat{\varphi}}(\psi \wedge \psi')$. By the induction hypothesis, it must be the case that $D_{\hat{\varphi}}(\psi \wedge \psi') = 0$, so (2.3) is vacuously true. Similarly for the case $\psi' \wedge \psi$. This shows that $\hat{\varphi}$ is free for $\neg$ on all depths > 0, and thus φ is equivalent to $\hat{\varphi}$ which is equivalent to $\partial\hat{\varphi}$ by Lemma 8 which is without weak negation. ■

Chapter 3

On the Relationship between Playing Rationally and Knowing how to Play: A Logical Account

Abstract. Modal logics of strategic ability usually focus on capturing what it means for an agent to have a feasible strategy that brings about some property. While there is a general agreement on abilities in scenarios where agents have perfect information, the right semantics for ability under incomplete information is still debated upon. Epistemic Temporal Strategic Logic, an offspring of this debate, can be treated as a logic that captures properties of agents' rational play. In this paper, we provide a semantics of ETSL that is more compact and comprehensible than the one presented in the original paper by van Otterloo and Jonker. Second, we use ETSL to show that a rational player knows that he will succeed if, and only if, he knows how to play to succeed – while the same is not true for rational coalitions of players.

Keywords: multi-agent systems, theories of agency, game-theoretical foundations, modal logic.

3.1 Introduction

Modal logics of strategic ability usually focus on capturing what it means for an agent to have a feasible strategy that brings about some property. While there is a general agreement on abilities in scenarios where agents have perfect information, the right semantics for ability under incomplete information is still debated upon. Epistemic Temporal Strategic Logic, proposed by van Otterloo and Jonker [135], is an offspring of this debate, but one that

leads in an orthogonal direction to the mainstream solutions. The central operator of ETSL can be read as: "if A play *rationally* to achieve φ (meaning: they never play a dominated strategy), they will achieve φ". Thus, one may treat ETSL as a logic that captures properties of agents' rational play in a sense.

This paper contains two main messages. First, we provide a semantics of ETSL that is more compact and comprehensible than the one presented in [135]. ETSL is underpinned by several exciting concepts. Unfortunately, its semantics is also quite hard to read due to a couple non-standard solutions and a plethora of auxiliary functions, which is probably why the logic never received the attention it deserves. Second, and perhaps more importantly, we use ETSL to show that a rational player knows that he will succeed if, and only if, he knows how to play to succeed – while the same is not true for rational coalitions of players.

3.2 Reasoning about Abilities of Agents

Modal logics of strategic ability [6, 8] form one of the fields where logic and game theory can successfully meet. The logics have clear possible worlds semantics, are axiomatizable, and have some interesting computational properties. Moreover, they are underpinned by intuitively appealing conceptual machinery for modeling and reasoning about systems that involve multiple autonomous agents.

3.2.1 ATL: Ability in Perfect Information Games

Alternating-time Temporal Logic (ATL) [6, 8] can be seen as a logic for systems involving multiple agents, that allows one to reason about what agents can achieve in game-like scenarios. Since ATL does not include incomplete information in its scope, it can be seen as a logic for reasoning about agents who always have perfect information about the current state of affairs. Formula $\langle\!\langle A\rangle\!\rangle\varphi$, where A is a coalition of agents, expresses that A have a collective strategy to enforce φ. ATL formulae include temporal operators: "$\bigcirc$" ("in the next state"), $\Box$ ("always from now on") and $\mathcal{U}$ ("until"). Operator $\Diamond$ ("now or sometime in the future") can be defined as $\Diamond\varphi \equiv \top\,\mathcal{U}\,\varphi$. Like in CTL, every occurrence of a temporal operator is preceded by exactly one cooperation modality $\langle\!\langle A\rangle\!\rangle$.[1] Formally, the recursive definition of ATL formulae is:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A\rangle\!\rangle \bigcirc \varphi \mid \langle\!\langle A\rangle\!\rangle \Box \varphi \mid \langle\!\langle A\rangle\!\rangle \varphi\, \mathcal{U}\, \varphi$$

A number of semantics have been defined for ATL, most of them equivalent [52]. In this paper, we use a variant of *concurrent game structures*,

$$M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o\rangle,$$

which includes a nonempty finite set of all agents $\mathbb{A}\mathrm{gt} = \{1, ..., k\}$, a nonempty set of states St, a set of atomic propositions Π, a valuation of propositions

[1] The logic to which such a syntactic restriction applies is sometimes called *"vanilla"* ATL (resp. "vanilla" CTL etc.).

$\pi : \Pi \to 2^{St}$, and a nonempty set of (atomic) actions Act. Function $d : \mathbb{A}\mathrm{gt} \times St \to 2^{Act}$ defines actions available to an agent in a state, and o is a deterministic transition function that assigns an outcome state $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to state q, and a tuple of actions $\langle \alpha_1, \ldots, \alpha_k \rangle$ that can be executed by $\mathbb{A}\mathrm{gt}$ in q. A *strategy* of agent a is a conditional plan that specifies what a is going to do for every possible situation ($s_a : St \to Act$ such that $s_a(q) \in d(a, q)$). A *collective strategy* (called also a *strategy profile*) S_A for a group of agents A is a tuple of strategies S_a, one per agent $a \in A$. A *path* λ in M is an infinite sequence of states that can be effected by subsequent transitions, and refers to a possible course of action (or a possible computation) that may occur in the system; by $\lambda[i]$, we denote the ith position on path λ. Function $out(q, S_A)$ returns the set of all paths that may result from agents A executing strategy S_A from state q onward:

$out(q, S_A) = \{\lambda = q_0 q_1 q_2 \ldots \mid q_0 = q$ and for every $i = 1, 2, \ldots$ there exists a tuple of actions $\langle \alpha_1^{i-1}, \ldots, \alpha_k^{i-1} \rangle$ such that $\alpha_a^{i-1} = S_a(q_{i-1})$ for each $a \in A$, $\alpha_a^{i-1} \in d(a, q_{i-1})$ for each $a \notin A$, and $o(q_{i-1}, \alpha_1^{i-1}, \ldots, \alpha_k^{i-1}) = q_i\}$.

Now, the semantics of ATL formulae can be given via the following clauses:

$M, q \models p$	iff $q \in \pi(p)$ (where $p \in \Pi$);
$M, q \models \neg\varphi$	iff $M, q \not\models \varphi$;
$M, q \models \varphi \wedge \psi$	iff $M, q \models \varphi$ and $M, q \models \psi$;
$M, q \models \langle\langle A \rangle\rangle \bigcirc \varphi$	iff there is a collective strategy S_A such that, for every $\lambda \in out(q, S_A)$, we have $M, \lambda[1] \models \varphi$;
$M, q \models \langle\langle A \rangle\rangle \Box \varphi$	iff there exists S_A such that, for every $\lambda \in out(q, S_A)$, we have $M, \lambda[i]$ for every $i \geq 0$;
$M, q \models \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$	iff there is S_A st. for every $\lambda \in out(q, S_A)$ there is $i \geq 0$, for which $M, \lambda[i] \models \psi$, and $M, \lambda[j] \models \varphi$ for every $0 \leq j < i$.

3.2.2 Strategic Ability and Incomplete Information

ATL is unrealistic in a sense: real-life agents seldom possess complete information about the current state of the world. *Alternating-time Temporal Epistemic Logic* (ATEL) [133] enriches the picture with an epistemic component, adding to ATL operators for representing agents' knowledge: $K_a\varphi$ reads as "agent a knows that φ". Additional operators $E_A\varphi$, $C_A\varphi$, and $D_A\varphi$ refer to *mutual knowledge* ("everybody knows"), *common knowledge*, and *distributed knowledge* among the agents from A. Models for ATEL extend concurrent game structures with epistemic accessibility relations $\sim_1, \ldots, \sim_k \subseteq Q \times Q$ (one per agent) for modeling agents' uncertainty; the relations are assumed to be equivalences. We will call such models *concurrent epistemic game structures* (CEGS) in the rest of the paper. Agent a's epistemic relation is meant to encode a's inability to distinguish between the (global) system states: $q \sim_a q'$ means that, while the system is in state q, agent a cannot determine whether it is not in q'. Then:

$M, q \models K_a\varphi$ iff φ holds for every q' such that $q \sim_a q'$.

Relations $\sim_A^E$, $\sim_A^C$ and $\sim_A^D$, used to model group epistemics, are derived from the individual relations of agents from A. First, $\sim_A^E$ is the union of relations $\sim_a$, $a \in A$. Next, $\sim_A^C$ is defined as the transitive closure of $\sim_A^E$. Finally,

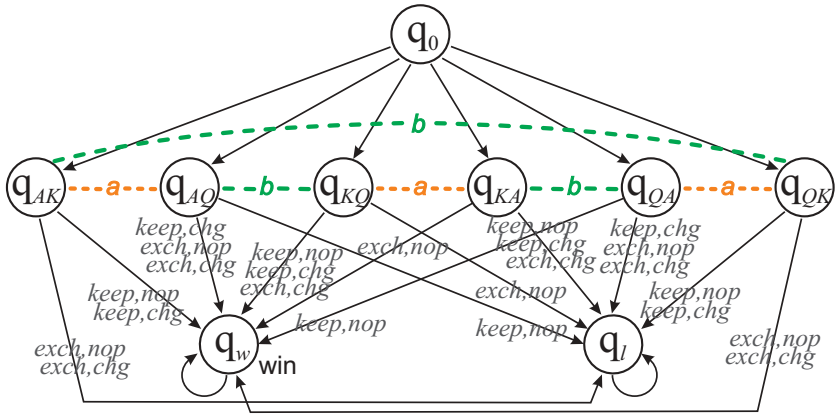


Figure 3.1: Gambling Robots game. Arrows represent possible transitions of the system (labeled with tuples of agents' actions); dashed lines connect states that are indiscernible for particular agents.

$\sim_A^D$ is the intersection of all the $\sim_a$, $a \in A$. The semantics of group knowledge can be defined as below (for $\mathcal{K} = C, E, D$):

$M, q \models \mathcal{K}_A\varphi$ iff φ holds for every q' such that $q \sim_A^{\mathcal{K}} q'$.

Example 12 (Gambling Robots) *Two robots (a and b) play a simple card game. The deck consists of Ace, King and Queen (A, K, Q); it is assumed that A beats K, K beats Q, but Q beats A. First, the "environment" agent env deals a random card to both robots (face down), so that each player can see his own hand, but he does not know the card of the other player. Then robot a can exchange his card for the one remaining in the deck (action $exch$), or he can keep the current one ($keep$). At the same time, robot b can change the priorities of the cards, so that A becomes better than Q (action chg) or he can do nothing (nop). If a has a better card than b after that, then a win is scored, otherwise the game ends in a "losing" state. A CEGS for the game is shown in Figure 3.1; we will refer to the model as M_0 throughout the rest of the paper. Note that $M_0, q_0 \models \langle\langle a\rangle\rangle\Diamond\mathsf{win}$ (and even $M_0, q_0 \models K_a\langle\langle a\rangle\rangle\Diamond\mathsf{win}$), although, intuitively, a has no feasible way of ensuring a win. This is a fundamental problem with ATEL, which we discuss briefly below.*

It was pointed out in several places that the meaning of ATEL formulae is somewhat counterintuitive [66, 83, 86]. Most importantly, one would expect that an agent's ability to achieve property φ should imply that the agent has enough control and knowledge to *identify* and *execute* a strategy that enforces φ (cf. also [121]). This problem is closely related to the well known distinction between knowledge *de re* and knowledge *de dicto*.

A number of frameworks were proposed to overcome this problem [66, 83, 121, 86, 135, 63], yet none of them seems the ultimate definitive solution. Most of the solutions agree that only *uniform* strategies (i.e., strategies that specify the same choices in indistinguishable states) are really executable. However, in order to identify a successful strategy, the agents must consider not only the courses of action, starting from the current state of the

system, but also from states that are indistinguishable from the current one. There are many cases here, especially when group epistemics is concerned: the agents may have common, ordinary or distributed knowledge about a strategy being successful, or they may be hinted the right strategy by a distinguished member (the "boss"), a subgroup ("headquarters committee") or even another group of agents ("consulting company"). Most existing solutions [121, 135, 63] treat only some of the cases (albeit rather in an elegant way), while others [83, 86] offer a more general treatment of the problem at the expense of an overblown logical language (which is by no means elegant).

Recently, a new, non-standard semantics for ability under incomplete information has been proposed in [73, 75], which we believe to be both intuitive, general and elegant. We summarize the proposal in the next section, as we will use it further to capture strategic abilities of agents.

3.2.3 An Intuitive Semantics for Ability and Knowledge

In [73, 75], a non-standard semantics for the logic of strategic ability and incomplete information has been proposed, which we believe to be finally satisfying. In the semantics, formulae are interpreted over *sets of states* rather than single states. Moreover, we introduce "constructive knowledge" operators $\mathbb{K}_a$, one for each agent a, that yield the set of states, indistinguishable from the current state from a's perspective. Constructive common, mutual, and distributed knowledge is formalized via operators $\mathbb{C}_A, \mathbb{E}_A$, and $\mathbb{D}_A$. The language, which we tentatively call Constructive Strategic Logic (CSL) here, is defined as follows:

$$\varphi ::= p \mid \neg\varphi \mid {\sim}\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle \bigcirc \varphi \mid \langle\langle A\rangle\rangle \Box \varphi \mid \langle\langle A\rangle\rangle \varphi \mathcal{U} \varphi \mid C_A\varphi \mid E_A\varphi \mid D_A\varphi \mid \mathbb{C}_A\varphi \mid \mathbb{E}_A\varphi \mid \mathbb{D}_A\varphi.$$

Individual knowledge operators can be derived as: $K_a\varphi \equiv E_{\{a\}}\varphi$ and $\mathbb{K}_a\varphi \equiv \mathbb{E}_{\{a\}}\varphi$. Moreover, we define $\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, and $\varphi_1 \rightarrow \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2$.

The models are concurrent epistemic game structures again, and we consider only memoryless uniform strategies. Let $\mathrm{img}(q, \mathcal{R})$ be the image of state q with respect to relation $\mathcal{R}$, i.e. the set of all states q' such that $q\mathcal{R}q'$. Moreover, we use $out(Q, S_A)$ as a shorthand for $\cup_{q\in Q} out(q, S_A)$, and $\mathrm{img}(Q, \mathcal{R})$ as a shorthand for $\cup_{q\in Q}\mathrm{img}(q, \mathcal{R})$. The notion of a formula φ being satisfied by a set of states $Q \subseteq St$ in a model M is given through the following clauses.

$M, Q \models p$	iff $q \in \pi(p)$ for every $q \in Q$;
$M, Q \models \neg\varphi$	iff $M, Q \not\models \varphi$;
$M, Q \models {\sim}\varphi$	iff $M, q \not\models \varphi$ for every $q \in Q$;
$M, Q \models \varphi \wedge \psi$	iff $M, Q \models \varphi$ and $M, Q \models \psi$;
$M, Q \models \langle\langle A\rangle\rangle \bigcirc \varphi$	iff there exists S_A such that, for every $\lambda \in out(Q, S_A)$, we have that $M, \{\lambda[1]\} \models \varphi$;
$M, Q \models \langle\langle A\rangle\rangle \Box \varphi$	iff there exists S_A such that, for every $\lambda \in out(Q, S_A)$ and $i \geq 0$, we have $M, \{\lambda[i]\} \models \varphi$;

$M, Q \models \langle\langle A\rangle\rangle \varphi \mathcal{U} \psi$ iff there exists S_A such that, for every $\lambda \in out(Q, S_A)$, there is $i \geq 0$ for which $M, \{\lambda[i]\} \models \psi$ and $M, \{\lambda[j]\} \models \varphi$ for every $0 \leq j < i$;

$M, Q \models \mathcal{K}_A \varphi$ iff $M, q \models \varphi$ for every $q \in \mathrm{img}(Q, \sim_A^{\mathcal{K}})$ (where $\mathcal{K} = C, E, D$);

$M, Q \models \hat{\mathcal{K}}_A \varphi$ iff $M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}) \models \varphi$ (where $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$ and $\mathcal{K} = C, E, D$, respectively).

We will also write $M, q \models \varphi$ as a shorthand for $M, \{q\} \models \varphi$, and this is the notion of satisfaction (in single states) that we are ultimately interested in – but that notion is defined in terms of the satisfaction in sets of states.

Now, $\mathbb{K}_a \langle\langle a\rangle\rangle \varphi$ expresses the fact that a has a single strategy that enforces φ from *all* states indiscernible from the current state, instead of stating that φ can be achieved from *every* such state *separately* (what $K_a \langle\langle a\rangle\rangle \varphi$ says, which is very much in the spirit of standard epistemic logic). More generally, the first kind of formulae refer to *having a strategy "de re"* (i.e. having a successful strategy and knowing the strategy), while the latter refer to *having a strategy "de dicto"* (i.e. only knowing that *some* successful strategy is available; cf. [83]). Note also that the property of having a winning strategy in the current state (but not necessarily even knowing *about* it) is simply expressed with $\langle\langle a\rangle\rangle \varphi$. Capturing different ability levels of coalitions is analogous, with various "epistemic modes" of collective recognizing the right strategy.

Example 13 *Robot a has no winning strategy in the starting state of the game: $M_0, q_0 \models \neg\langle\langle a\rangle\rangle \Diamond \mathsf{win}$, which implies that it has neither a strategy "de re" nor "de dicto" ($M_0, q_0 \models \neg\mathbb{K}_a \langle\langle a\rangle\rangle \Diamond \mathsf{win} \wedge \neg K_a \langle\langle a\rangle\rangle \Diamond \mathsf{win}$). On the other hand, he has a successful strategy in q_{AK} (just play $keep$) and he knows he has one (because another action, $exch$, is bound to win in q_{AQ}); still, the knowledge is not constructive, since a does not know which strategy is the right one in the current situation: $M_0, q_{AK} \models \langle\langle a\rangle\rangle \bigcirc \mathsf{win} \wedge K_a \langle\langle a\rangle\rangle \bigcirc \mathsf{win} \wedge \neg\mathbb{K}_a \langle\langle a\rangle\rangle \bigcirc \mathsf{win}$. Also, b's playing chg enforces a transition to q_w for both q_{AQ}, q_{KQ}, so $M_0, q_{AQ} \models \mathbb{K}_b \langle\langle b\rangle\rangle \bigcirc \mathsf{win}$ (robot b has a strategy "de re" to enforce a win from q_{AQ}).*

Finally, $q_{QK} \models \langle\langle a, b\rangle\rangle \Diamond \mathsf{win} \wedge E_{\{a,b\}} \langle\langle a, b\rangle\rangle \Diamond \mathsf{win} \wedge C_{\{a,b\}} \langle\langle a, b\rangle\rangle \Diamond \mathsf{win} \wedge \neg\mathbb{E}_{\{a,b\}} \langle\langle a, b\rangle\rangle \Diamond \mathsf{win} \wedge \mathbb{D}_{\{a,b\}} \langle\langle a, b\rangle\rangle \Diamond \mathsf{win}$: in q_{QK}, the robots have a collective strategy to enforce a win, and they all know it (they even have common knowledge about it); on the other hand, they cannot identify the right strategy as a team – they can only see one if they share knowledge at the beginning (i.e., in q_{QK}).

3.3 Epistemic Temporal Strategic Logic

A very interesting variation on the theme of combining strategic, epistemic and temporal aspects of a multi-agent system was proposed in [135]. Epistemic Temporal Strategic Logic (ETSL) digs deeper in the repository of game theory, and focuses on the concept of *undominated strategies*. Thus, its variant of cooperation modalities has a different flavor than the ones from ATL, ATEL, CSL etc. In a way, formula $\langle\langle A\rangle\rangle \varphi$ in ETSL can be summarized as:

"If A play *rationally* to achieve φ (meaning: they never play a dominated strategy), they will achieve φ".

ETSL can be treated as a logic that describes the outcome of *rational play* under incomplete information,[2] in the same way as CSL can be seen as a logic that captures agents' strategic abilities (regardless of whether the agents play rationally or not). The main claim we propose in this paper is that a rational player knows that he will succeed if, and only if, he has a strategy "de re" to succeed – while the same is not true for rational coalitions of players. However, before we present and discuss the claim formally in Section 3.4, we must re-write the semantics of ETSL in several respects.

First, the original semantics of ETSL is defined only for finite turn-based acyclic game models with epistemic accessibility relations, and we will generalize the semantics to concurrent epistemic game structures. Next, the semantics comes with a plethora of auxiliary functions and definitions (and a couple of omissions), which makes it rather hard to read. In fact, this is probably the reason why the logic never received the attention it deserves, and it is definitely worth trying to make the semantics more compact. Finally, the authors of [135] propose that a model should include also a "grand strategy profile" $S_{\mathbb{A}\mathrm{gt}}$, defining the actual strategies of all agents (or at least constraining them in some way, since non-deterministic strategies are allowed in ETSL). While the idea seems interesting in itself (a similar idea was later exploited e.g. in [85] to allow for explicit analysis of strategies and reasoning about strategy revision), we will show that it does not introduce a finer-grained analysis of "vanilla" ETSL formulas: if a formula holds in M, q for one strategy profile, it holds in M, q for all the other strategy profiles, too. Moreover, it can be proved that the semantics of cooperation modalities $\langle\langle A\rangle\rangle$ is the same regardless of whether we consider non-deterministic strategies or not. In consequence, we will be able to show a "vanilla" ETSL semantics expressed entirely in terms of concurrent epistemic game structures and their states.

3.3.1 The Semantics Made Easier to Read

Formulae of ETSL come with no restriction wrt grouping of temporal operators:

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle\langle A\rangle\rangle\varphi \mid \bigcirc\varphi \mid \Box\varphi \mid \varphi\,\mathcal{U}\,\psi \mid K_a\varphi.$$

After some re-writing (and having it generalized to general game structures, not only turn-based trees), the semantics can be given as follows. Strategies are allowed to be non-deterministic, i.e. $S_a : St \to 2^{Act}$.[3] We require strategies to be uniform, although [135] does not do it explicitly (we take it as a simple omission, because otherwise many claims in that paper seem to be false). A collective strategy (strategy profile) S_A is a tuple of strategies, one per agent from A. S^0_a is the "neutral strategy" with no restriction on a's actions ($S^0_a(q) = Act$ for each $q \in St$), and strategy profile S^0_A assigns neutral strategies to agents from A. Moreover, we generalize function $out(q, S_A)$ to handle nondeterministic strategies too; in $out'(q, S_A)$, "$\alpha^{i-1}_a = S_a(q_{i-1})$" is replaced with $\alpha^{i-1}_a \in S_a(q_{i-1})$.

[2] We emphasize that this is a specific notion of rationality (i.e., agents are assumed to *play only undominated strategies*). Game theory proposes several other rationality criteria as well, based e.g. on Nash equilibrium, dominant strategies, or Pareto efficiency. In fact, it is easy to imagine ETSL-like logics based on these notions instead.

[3] To preserve seriality ("time flows forever"), we assume that $S_a(q) \neq \varnothing$ for all $q \in St$.

Now, the semantics can be given through the following clauses (the semantics for p, $\neg\varphi$ and $\varphi \wedge \psi$ is analogous to the one presented in Section 3.2.1):

$M, S_{\mathbb{A}gt}, q \models \langle\langle A\rangle\rangle\varphi$ iff for all strategies T_A, undominated wrt q, φ, we have $M, (T_A, S^0_{\mathbb{A}gt\setminus A}), q \models \varphi$;

$M, S_{\mathbb{A}gt}, q \models \bigcirc\varphi$ iff for every $\lambda \in out'(q, S_{\mathbb{A}gt})$ we have $M, S_{\mathbb{A}gt}, \lambda[1] \models \varphi$;

$M, S_{\mathbb{A}gt}, q \models \Box\varphi$ iff for every $\lambda \in out'(q, S_{\mathbb{A}gt})$ and $i \geq 0$ we have $M, S_{\mathbb{A}gt}, \lambda[i] \models \varphi$;

$M, S_{\mathbb{A}gt}, q \models \varphi\,\mathcal{U}\,\psi$ iff for every $\lambda \in out'(q, S_{\mathbb{A}gt})$ there is $i \geq 0$ such that $M, S_{\mathbb{A}gt}, \lambda[i] \models \psi$ and for all j such that $0 \leq j < i$ we have $M, S_{\mathbb{A}gt}, \lambda[j] \models \varphi$;

$M, S_{\mathbb{A}gt}, q \models K_a\varphi$ iff for all $q \sim_a q'$ we have $M, (S_{\mathbb{A}gt}(a), S^0_{\mathbb{A}gt\setminus\{a\}}), q' \models \varphi$.

Definition 6 *Strategy S_A* dominates *T_A with respect to formula φ, model M, and state q, if S_A achieves φ better then T_A, i.e. iff:*

1. *for every q' such that $q \sim_A q'$: if $M, (T_A, S^0_{\mathbb{A}gt\setminus A}), q' \models \varphi$ then also $M, (S_A, S^0_{\mathbb{A}gt\setminus A}), q' \models \varphi$, and*
2. *there exists q' such that $q \sim_A q'$, and $M, (S_A, S^0_{\mathbb{A}gt\setminus A}), q' \models \varphi$, and $M, (T_A, S^0_{\mathbb{A}gt\setminus A}), q \not\models \varphi$.*

Remark 8 *Definition 6 uses epistemic relation $\sim_A$. However, epistemic accessibility relations are defined only for individual agents in [135], which is perhaps another omission. In this study, we take the liberty to fix $\sim_A$ as $\sim^E_A$.*

We also point out that ETSL can be extended with collective epistemic operators E_A, C_A, D_A in a straightforward manner.

Example 14 *Consider the gambling robots again. Robot a has two undominated strategies wrt $\bigcirc\mathsf{win}$, M, q_{AK}: namely, to play $exch$ in both q_{AK}, q_{AQ}, or to play $keep$ in both (other choices do not matter). Since playing $exch$ fails in q_{AK}, so: $M_0, q_{AK} \not\models \langle\langle a\rangle\rangle\bigcirc\mathsf{win}$. Furthermore, playing $keep$ is the only undominated strategy in q_{KQ} and q_{KA} (and it succeeds only in q_{KQ}). Thus, $M_0, q_{KQ} \models \langle\langle a\rangle\rangle\bigcirc\mathsf{win}$, and $M_0, q_{KA} \not\models \langle\langle a\rangle\rangle\bigcirc\mathsf{win}$. Hence, $M_0, q_{KQ} \not\models K_a\langle\langle a\rangle\rangle\bigcirc\mathsf{win}$.*

3.3.2 A Few Properties

In this section, we present several properties of ETSL formulae that will allow us to give an even simpler semantic definition of "vanilla" ETSL.

Proposition 29 *For every "vanilla" ETSL formula φ, concurrent epistemic game structure M, and state q in M: $M, S_{\mathbb{A}gt}, q \models \varphi$ iff $M, S'_{\mathbb{A}gt}, q \models \varphi$ for any pair of "grand" strategy profiles $S_{\mathbb{A}gt}, S'_{\mathbb{A}gt}$.*

Proof By induction on the structure of φ. Note that it is sufficient to prove the implication one way, as the choice of $S_{\mathbb{A}gt}, S'_{\mathbb{A}gt}$ is completely arbitrary.

Case $\varphi \equiv p$**:** $M, S_{\mathbb{A}gt}, q \models p$, so $q \in \pi(q)$, so $M, S'_{\mathbb{A}gt}, q \models p$.

Case $\varphi \equiv \neg\psi$**:** $M, S_{\mathbb{A}gt}, q \models \neg\psi$, so $M, S_{\mathbb{A}gt}, q \not\models \psi$, so (by induction hypothesis) $M, S'_{\mathbb{A}gt}, q \not\models \psi$, so $M, S'_{\mathbb{A}gt}, q \models \neg\psi$. (As the choice of $S_{\mathbb{A}gt}, S'_{\mathbb{A}gt}$ was completely arbitrary, the implication holds the other way too.)

Case $\varphi \equiv \psi_1 \wedge \psi_2$**:** analogous.

Case $\varphi \equiv \langle\langle A\rangle\rangle \bigcirc \psi$**:** $M, S_{\mathbb{A}\mathrm{gt}}, q \models \langle\langle A\rangle\rangle \bigcirc \psi$ iff $M, (T_A, S^0_{\mathbb{A}\mathrm{gt}\setminus A}), \lambda[1] \models \varphi$ for all undominated T_A and $\lambda \in out'(q, (T_A, S^0_{\mathbb{A}\mathrm{gt}\setminus A}))$. Note that the latter condition does not refer to $S_{\mathbb{A}\mathrm{gt}}$, so $M, S'_{\mathbb{A}\mathrm{gt}}, q \models \langle\langle A\rangle\rangle \bigcirc \psi$ too.

Cases $\varphi \equiv \langle\langle A\rangle\rangle \Box \psi$ **and** $\varphi \equiv \langle\langle A\rangle\rangle \psi_1 \mathcal{U} \psi_2$**:** analogous.

Case $\varphi \equiv K_a\psi$**:** $M, S_{\mathbb{A}\mathrm{gt}}, q \models K_a\psi$, so $M, (S_{\mathbb{A}\mathrm{gt}}(a), S^0_{\mathbb{A}\mathrm{gt}\setminus\{a\}}), q' \models \psi$ for all $q \sim_a q'$. By induction hypothesis, also $M, (S'_{\mathbb{A}\mathrm{gt}}(a), S^0_{\mathbb{A}\mathrm{gt}\setminus\{a\}}), q' \models \psi$ for all $q \sim_a q'$, so $M, S'_{\mathbb{A}\mathrm{gt}}, q \models K_a\psi$.

■

Remark 9 *We point out that restricting the scope of Proposition 29 to "vanilla" ETSL formulae is important. In particular, the epistemic opertor K_a has a non-standard interpretation when the full language of ETSL is considered.*

Proposition 30 *Let $\Phi \equiv \bigcirc \psi, \Box\psi$, or $\psi_1 \mathcal{U} \psi_2$ where ψ, ψ_1, ψ_2 are "vanilla" ETSL formulae. Moreover, let $|\Phi|$ denote the set of paths for which Φ holds; formally, $|\bigcirc \psi| = \{\lambda \mid M, \lambda[1] \models \psi\}$, $|\Box\psi| = \{\lambda \mid \forall_i M, \lambda[i] \models \psi\}$, and $|\psi_1 \mathcal{U} \psi_2| = \{\lambda \mid \exists_i (M, \lambda[i] \models \psi_2 \wedge \forall_{0 \leq j < i} M, \lambda[j] \models \psi_1\}$.*

Then, S_A dominates T_A wrt Φ, M, and q iff:

1. *for every q', $q \sim^E_A q'$: if $out(q', T_A) \subseteq |\Phi|$ then also $out(q', S_A) \subseteq |\Phi|$, and*
2. *there exists q', $q \sim^E_A q'$, such that $out(q', S_A) \subseteq |\Phi|$ and $out(q', T_A) \not\subseteq |\Phi|$.*

Proof Straightforward from the definition. ■

Remark 10 *Note that dominance can be characterized in an even more compact way. Let $succ_{q,\Phi}(S_A) = \{q \in \mathrm{img}(q, \sim^E_A) \mid out(q, S_A) \subseteq |\Phi|\}$ be the set of states from $\mathrm{img}(q, \sim^E_A)$, for which s_a succeeds to enforce Φ. Now, S_A dominates T_A wrt Φ, M, q iff $succ_{q,\Phi}(T_A) \subsetneq succ_{q,\Phi}(S_A)$.*

Proposition 31 *Let $\Phi \equiv \bigcirc \psi, \Box\psi$, or $\psi_1 \mathcal{U} \psi_2$ where ψ, ψ_1, ψ_2 are "vanilla" ETSL formulae. Strategy T_A is dominated wrt Φ, M, q by a strategy S_A iff it is dominated wrt Φ, M, q by a* deterministic *strategy S'_A.*

Proof $\Rightarrow$: Let T_A be dominated by S_A (wrt φ, M, q). We construct the deterministic strategy S'_A by fixing arbitrary (uniform) choices out of S_A. Formally, for every agent $a \in A$ and abstraction class $\mathrm{img}(q', \sim_a) \subseteq St$ such that $S_a(q') = \{\alpha, \alpha', ...\}$, we fix $S'_a(q'') = \alpha$ for all $q'' \in \mathrm{img}(q', \sim_a)$. (By uniformity of S_A, we have $\alpha \in S_a(q'')$ for all $q'' \in \mathrm{img}(q', \sim_a)$, so S'_A is a valid strategy.) First, this enforces uniformity of S'_A. Second, $out(\bar{q}, S'_A) \subseteq out(\bar{q}, S_A)$ for all $\bar{q} \in St$ (by definition of out). Thus, we can use Proposition 30 to show that S'_A dominates T_A, which concludes the proof.

$\Leftarrow$: Straightforward. ■

Proposition 32 *Let Φ be as above. Then, $M, S_{\mathbb{A}\mathrm{gt}}, q \models \langle\langle A\rangle\rangle\Phi$ iff for all* deterministic *strategies T_A, undominated wrt Φ, we have $M, (T_A, S^0_{\mathbb{A}\mathrm{gt}\setminus A}), q \models \Phi$.*

Proof $\Rightarrow$: Straightforward.

$\Leftarrow$: Assume that $M, (T_A, S^0_{\mathbb{A}\mathrm{gt}\setminus A}), q \models \Phi$ for all deterministic strategies T_A, undominated wrt Φ, and suppose that there is a nondeterministic undominated S_A such that $M, (S_A, S^0_{\mathbb{A}\mathrm{gt}\setminus A}), q \not\models \Phi$. Let us fix a deterministic uniform strategy S'_A out of S_A in a similar way as in Proposition 31. Now, $out(\bar{q}, S'_A) \subseteq out(\bar{q}, S_A)$ for all $\bar{q} \in St$, so $out(q', S_A) \subseteq |\Phi|$ implies $out(q', S'_A) \subseteq |\Phi|$ (S'_A is never worse than S_A wrt Φ). Moreover, $out(q, S'_A) \subseteq |\Phi|$ and $out(q, S_A) \not\subseteq |\Phi|$. By Proposition 30, S'_A dominates S_A, so S_A is dominated – a contradiction. ■

3.3.3 ETSL in Terms of Concurrent Epistemic Game Structures

We have shown that, for "vanilla" ETSL, strategies do not have to be referred explicitly in the interpretation of formulae (Propositions 29 and 30). Moreover, we can restrict the set of considered strategies to deterministic strategies (Propositions 31 and 32). In consequence, we can express the semantics of "vanilla" ETSL equivalently in ATL-like fashion:

$M, q \models \langle\langle A\rangle\rangle\bigcirc\varphi$ iff for every strategy S_A, undominated wrt $q, \bigcirc\varphi$, and every $\lambda \in out(q, S_A)$, we have that $M, \lambda[1] \models \varphi$;

$M, q \models \langle\langle A\rangle\rangle\Box\varphi$ iff for every strategy S_A, undominated wrt $q, \Box\varphi$, and every $\lambda \in out(q, S_A)$ and $i \geq 0$ we have $M, \lambda[i] \models \varphi$;

$M, q \models \langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\psi$ iff for every strategy S_A, undominated wrt $q, \varphi\,\mathcal{U}\,\psi$, and every $\lambda \in out(q, S_A)$, there is $i \geq 0$ such that $M, \lambda[i] \models \psi$ and for all j such that $0 \leq j < i$ we have $M, \lambda[j] \models \varphi$.

Only uniform deterministic strategies are taken into account. The semantics of p, $\neg\varphi$, $\varphi \wedge \psi$, and the epistemic operators is the same as for ATL and ATEL.

3.4 Playing Rationally vs. Knowing how to Play

We can finally present the main result of this paper, namely, that a rational player knows that he will succeed if, and only if, he has a strategy "de re" to succeed. The result holds under the assumption that the model is finite,[4] or more generally, that it includes at least one undominated strategy.

Moreover, we show that having common knowledge how to succeed is, in general, a stronger property than knowing that one will succeed for rational coalitions of players. That is, if rational agents have common knowledge about a winning strategy, then they have common knowledge that they will succeed – but the converse is not true any more. Surprisingly enough, it turns out that the relationship is strictly reverse for distributed knowledge:

[4] We use the term "finite model" to denote a CEGS with a *finite set of states* St.

if a rational coalition has distributed knowledge that it will succeed, then it has distributed knowledge about a winning strategy – but not necessarily the other way around. For mutual knowledge, the relationship holds neither way.

In what follows, we use $\models_{\text{ETSL}}$ and $\models_{\text{CSL}}$ to denote the ETSL and CSL satisfaction relation, respectively.

3.4.1 Rational Play of Individual Agents

We begin with two important lemmas.

Lemma 9 *Given a finite model M, state q in M, formula Φ and agent a, there is a strategy s_a which is undominated wrt M, q, Φ.*

Proof First, we consider the simpler case when the set of actions Act is finite. In such a case, the set of strategies is also finite, and the dominance relation is transitive and antireflexive. Suppose that every strategy is dominated; then, there must be a strategy which is dominated by itself – a contradiction.

We sketch the proof for infinite Act as follows. We partition the infinite set of strategies into equivalence classes, such that strategies in the same class have the same outcome paths for every state q (i.e., $s_a \approx t_a$ iff $\forall_q out(q, s_a) = out(q, t_a)$). Obviously, if s_a dominates t_a, then all strategies $s'_a \approx s_a$ dominate t_a too. Now, at every state q (and therefore at every point on a path from $out(q', s_a)$) there is a finite number of possible sets of successor states (the actual set being determined by the choice $s_a(q)$). Moreover, the same choice must be taken at every further occurrence of the same state q on a path, since s_a is a memoryless strategy. In consequence, there is only a finite number of different sets of outcome paths, and hence a finite number of the equivalence classes. Again, dominance is transitive and antireflexive, so an undominated strategy must exist. ■

Remark 11 *Note that the result in Lemma 9 does not extend to CEGS with infinite state spaces. Consider the game of "Fuzzy Blackjack" (called so all the more because our robots play it usually after having consumed too much machine oil). Only a single player is necessary, and we use positive real numbers as states and actions (i.e., $St = Act = \mathbb{R}_+$). When the player chooses a number in state q, the number is added to the state: $o(q, \alpha) = q + \alpha$. The values below 1 are the winning ones, i.e. $\pi(\mathsf{win}) = (0, 1)$ (it should be 21, but this would make the game too complicated for a drunken robot). Moreover, the robot cannot distinguish between the states below 1: $q \sim_a q'$ for all $q, q' \in (0, 1)$. Now, there is no undominated strategy wrt $0.5, \bigcirc\mathsf{win}$.*

To prove this, suppose that a strategy s_a is undominated. The strategy is uniform, so $s_a(q) = \alpha$ for some $\alpha \in \mathbb{R}_+$ and all $q \in (0, 1)$. Obviously, $\alpha \in (0, 1)$, because else s_a never succeeds. Now, the set of states in which s_a is successful is: $succ_{0.5,\bigcirc\mathsf{win}}(s_a) = (0, 1 - \alpha)$. Let $t_a(q) = q + \alpha/2$. Now, $succ_{0.5,\bigcirc\mathsf{win}}(t_a) = (0, 1 - \alpha/2) \supsetneq succ_{0.5,\Phi}(s_a)$ – a contradiction. Note also that:

- *If we replace $\mathbb{R}_+$ with the set of positive rational numbers, the result is the same. So, there may be no undominated strategies even when we restrict St and Act to countable sets.*

- *In order to show the same for countable St and* finite *Act, it is sufficient to modify the example so that $Act = \{0, 1, call\}$, and the initial state and every subsequent action $\alpha = 0, 1$ are simply stored in the resulting state. Now $o(q, call)$ takes the initial state q_0 and the string of 0s and 1s $\alpha_1, ..., \alpha_n$ stored in q, and returns $q' = q_0 + (0.\alpha_1...\alpha_n 1)_2$. For such a game, there is no undominated strategy wrt $0.5, \Diamond$*win.

Lemma 10 *Given M, q, Φ, a, if there is an undominated strategy wrt M, q, Φ, then there is also an undominated strategy wrt M, q', Φ for every $q' \in \mathrm{img}(q, \sim_a)$.*

Proof Take any s_a undominated wrt M, q, Φ (*). Suppose now that s_a is dominated by some strategy t_a wrt another state $q' \in \mathrm{img}(q, \sim_a)$ (**).

1. By (*) and Prop. 30: $\forall_{q'' \in \mathrm{img}(q, \sim_a)} (out(q'', t_a) \subseteq |\Phi| \Rightarrow out(q'', s_a) \subseteq |\Phi|)$.
2. By (**) and Prop. 30: $\exists_{q'' \in \mathrm{img}(q', \sim_a)} (out(q'', t_a) \subseteq |\Phi| \wedge out(q'', s_a) \not\subseteq |\Phi|)$.

Moreover, $\mathrm{img}(q, \sim_a) = \mathrm{img}(q', \sim_a)$ because is $\sim_a$ is an equivalence relation – which gives a contradiction between (1) and (2). ■

Remark 12 *We note that Lemma 10 may hold even for indistinguishability relations that are not equivalences. In fact, it is sufficient to require that $\sim_a$ is transitive. In that case, $q' \in \mathrm{img}(q, \sim_a)$ and $q'' \in \mathrm{img}(q', \sim_a)$ implies that $q'' \in \mathrm{img}(q, \sim_a)$, and we also get the contradiction.*

We are ready to prove the main claim of this paper now.

Theorem 11 *Let us consider only finite models, and formulae $\Phi \equiv \bigcirc \psi, \Box \psi$, or $\psi_1 \mathcal{U} \psi_2$ where ψ, ψ_1, ψ_2 are "vanilla" ETSL formulae. An agent has a strategy "de re" to enforce Φ if, and only if, he knows that his rational play will bring about Φ. Formally, for every finite M and state q in M:*

$$M, q \models_{ETSL} K_a \langle\langle a \rangle\rangle \Phi \quad \textit{iff} \quad M, q \models_{CSL} \mathbb{K}_a \langle\langle a \rangle\rangle \Phi.$$

Proof Induction on the structure of Φ. We prove the theorem for the case $\Phi \equiv \Box \psi$. Other cases are analogous.

$\Rightarrow$: Let $M, q \models_{\text{ETSL}} K_a \langle\langle a \rangle\rangle \Box \psi$. Then, $\forall_{q' \in \mathrm{img}(q, \sim_a)} M, q' \models_{\text{ETSL}} \langle\langle a \rangle\rangle \Box \psi$, and hence $M, q \models_{\text{ETSL}} \langle\langle a \rangle\rangle \Box \psi$ in particular. By Lemmas 9 and 10, there is a strategy s_a, undominated wrt $M, q', \Box \psi$ for every $q' \in \mathrm{img}(q, \sim_a)$.
Then: $\forall_{q' \in \mathrm{img}(q, \sim_a)} \forall_{\lambda \in out(q', s_a)} \forall_i M, \lambda[i] \models_{\text{ETSL}} \Box \psi$. By the induction hypothesis, also $\forall_{q' \in \mathrm{img}(q, \sim_a)} \forall_{\lambda \in out(q', s_a)} \forall_i M, \lambda[i] \models_{\text{CSL}} \psi$. Thus, $\forall_{\lambda \in out(\mathrm{img}(q, \sim_a), s_a)} \forall_i M, \lambda[i] \models_{\text{CSL}} \psi$ and so $M, \mathrm{img}(q, \sim_a) \models_{\text{CSL}} \langle\langle a \rangle\rangle \Box \psi$, and finally $M, q \models_{\text{CSL}} \mathbb{K}_a \langle\langle a \rangle\rangle \Box \psi$.

$\Leftarrow$: Let $M, q \models_{\text{CSL}} \mathbb{K}_a \langle\langle a \rangle\rangle \Box \psi$, i.e. $M, \mathrm{img}(q, \sim_a) \models_{\text{CSL}} \langle\langle a \rangle\rangle \Box \psi$. Consider $q' \in \mathrm{img}(q, \sim_a)$. By transitivity of $\sim_a$, we have $\mathrm{img}(q', \sim_a) \subseteq \mathrm{img}(q, \sim_a)$, so also $\forall_{q' \in \mathrm{img}(q, \sim_a)} M, \mathrm{img}(q', \sim_a) \models_{\text{CSL}} \langle\langle a \rangle\rangle \Box \psi$. Then, for every $q' \in \mathrm{img}(q, \sim_a)$, there must be s_a such that $\forall_{q'' \in \mathrm{img}(q', \sim_a)} \forall_{\lambda \in out(q'', s_a)} \forall_i M, \lambda[i] \models_{\text{CSL}} \psi$, and hence (by induction) $\forall_{q'' \in \mathrm{img}(q', \sim_a)} \forall_{\lambda \in out(q'', s_a)} \forall_i M, \lambda[i] \models_{\text{ETSL}} \psi$. So, $succ_{q', \Box \psi}(s_a) = \mathrm{img}(q', \sim_a)$, and therefore $succ_{q', \Box \psi}(t_a) = \mathrm{img}(q', \sim_a)$ for every other undominated strategy t_a (otherwise t_a would be dominated by s_a). Thus, $M, q' \models_{\text{ETSL}} \langle\langle a \rangle\rangle \Box \psi$ for every $q' \in \mathrm{img}(q, \sim_a)$, and finally $M, q \models_{\text{ETSL}} K_a \langle\langle a \rangle\rangle \Box \psi$. ■

Theorem 12 *More generally, for every Φ as above, and M, q such that there exists an undominated strategy wrt M, q, Φ: $M, q \models_{ETSL} K_a\langle\langle a\rangle\rangle\Phi$ iff $M, q \models_{CSL} \mathbb{K}_a\langle\langle a\rangle\rangle\Phi$.*

3.4.2 Rational Coalitions Are at Disadvantage

Beside some philosophical insight into the nature of knowledge and rational play, Theorems 11 and 12 provide us with an alternative way of decomposing strategic abilities under incomplete information into a strategic and epistemic part. The definition of the strategic dimension is more sophisticated and less straightforward than usually; on the other hand, we do not pay the price of a non-standard satisfaction relation. Unfortunately, such decomposition is not valid any more when abilities of collective agents are concerned. Now, the relationship is much more limited: if a coalition has *common* knowledge how to play, then it has also common knowledge that rational play will be successful; the same does *not* hold for other types of collective knowledge. Moreover, the converse relationship is guaranteed for distributed knowledge, but *not* for common nor mutual knowledge.

Theorem 13 *Let $\Phi \equiv \bigcirc\psi, \Box\psi$, or $\psi_1 \mathcal{U} \psi_2$ where ψ, ψ_1, ψ_2 are "vanilla" ETSL formulae. Then, if a coalition has common knowledge how to play, then it has common knowledge that rational play will be successful:*

$$\textit{if} \quad M, q \models_{CSL} \mathbb{C}_A\langle\langle A\rangle\rangle\Phi \quad \textit{then} \quad M, q \models_{ETSL} C_A\langle\langle A\rangle\rangle\Phi.$$

The same holds for neither mutual nor distributed knowledge.

Proof **Common knowledge:** Let $M, q \models_{\text{CSL}} \mathbb{K}_A\langle\langle A\rangle\rangle\Box\psi$, i.e. $M, \text{img}(q, \sim_A^C) \models_{\text{CSL}} \langle\langle A\rangle\rangle\Box\psi$. Consider $q' \in \text{img}(q, \sim_A^C)$. We have $\text{img}(q', \sim_A^E) \subseteq \text{img}(q', \sim_A^C) \subseteq \text{img}(q, \sim_A^C)$, so also $\forall_{q' \in \text{img}(q, \sim_A^C)} M, \text{img}(q', \sim_A^E) \models_{\text{CSL}} \langle\langle A\rangle\rangle\Box\psi$. Then, for every $q' \in \text{img}(q, \sim_A^C)$, there must be S_A such that $\forall_{q'' \in \text{img}(q', \sim_A^E)} \forall_{\lambda \in out(q'', S_A)} \forall_i M, \lambda[i] \models_{\text{CSL}} \psi$, and hence (by induction) $\forall_{q'' \in \text{img}(q', \sim_A^E)} \forall_{\lambda \in out(q'', S_A)} \forall_i M, \lambda[i] \models_{\text{ETSL}} \psi$. So, $succ_{q', \Box\psi}(S_A) = \text{img}(q', \sim_A^E)$, and therefore $succ_{q', \Box\psi}(T_A) = \text{img}(q', \sim_A^E)$ for every other undominated strategy T_A (otherwise T_A would be dominated by S_A). Thus, $M, q' \models_{\text{ETSL}} \langle\langle A\rangle\rangle\Box\psi$ for every $q' \in \text{img}(q, \sim_A^C)$, and finally $M, q \models_{\text{ETSL}} C_A\langle\langle A\rangle\rangle\Box\psi$.

Mutual knowledge: for a counterexample, consider a modification of the game from Figure 3.1, in which a third robot c is introduced. The robot can only execute nop, and its epistemic relation $\sim_c = \{(q, q) \mid q \in St\} \cup \{(q_{KQ}, q_{KA}), (q_{KA}, q_{KQ})\}$, i.e. c can distinguish all states except q_{KQ}, q_{KA}. Moreover, the transition function is slightly changed: now, $o(q_{KA}, keep, nop) = q_w$. For the resulting system M_1, we have that $M_1, q_{AQ} \models_{\text{CSL}} \mathbb{E}_{\{b,c\}}\langle\langle b, c\rangle\rangle\bigcirc \text{win}$, but at the same time $M_1, q_{AQ} \not\models_{\text{ETSL}} E_{\{a,c\}}\langle\langle a, c\rangle\rangle\bigcirc \text{win}$ because $M_1, q_{KQ} \not\models_{\text{ETSL}} \langle\langle a, c\rangle\rangle\bigcirc \text{win}$.

Distributed knowledge: analogously, $M_1, q_{KQ} \models_{\text{CSL}} \mathbb{D}_{\{b,c\}}\langle\langle b, c\rangle\rangle\bigcirc \text{win}$, yet $M_1, q_{KQ} \not\models_{\text{ETSL}} D_{\{a,c\}}\langle\langle a, c\rangle\rangle\bigcirc \text{win}$ because $M_1, q_{KQ} \not\models_{\text{ETSL}} \langle\langle a, c\rangle\rangle\bigcirc \text{win}$. ■

Theorem 14 *Let $\Phi \equiv \bigcirc\psi, \Box\psi$, or $\psi_1 \mathcal{U} \psi_2$ where ψ, ψ_1, ψ_2 are "vanilla" ETSL formulae, and let M be a finite CEGS.*[5] *Then, if A have distributed knowledge that*

[5] Alternatively, we can request that A have at least one undominated strategy for every relevant state.

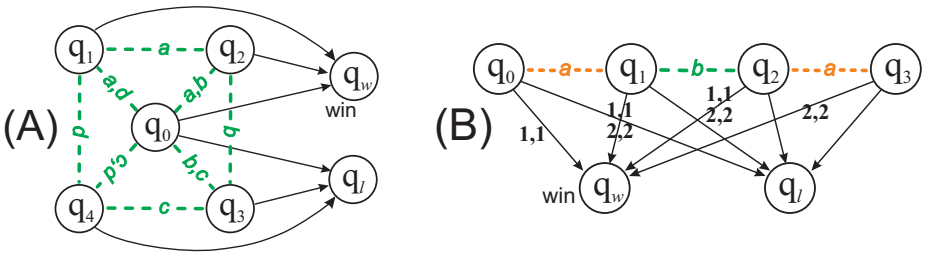


Figure 3.2: **(A)** Model M_2: four agents a, b, c, d, epistemic relations shown with the dashed lines, $Act = \{1, 2, 3, 4\}$. Transitions: $o(q_i, j, j, j, j) = q_w$ for $j \neq i$, otherwise the system proceeds to the "losing" state q_l; **(B)** Model M_3: two agents a, b, two actions $1, 2$. The tuples of actions that are absent in the graph lead to q_l.

rational play will bring about Φ, then they have distributed knowledge how to play to bring about Φ. Formally:

$$\textit{if} \quad M, q \models_{\text{ETSL}} D_A\langle\langle A\rangle\rangle\Phi \quad \textit{then} \quad M, q \models_{\text{CSL}} \mathbb{D}_A\langle\langle A\rangle\rangle\Phi.$$

The same holds for neither mutual nor common knowledge.

Proof sketch **Distributed knowledge:** the proof is analogous to the proofs of Lemma 10 and Theorem 11 (part $\Rightarrow$), as we can exploit the fact that $\sim_A^D$ is transitive, and $\text{img}(q, \sim_A^D) \subseteq \text{img}(q, \sim_A^E)$.

Mutual knowledge: for a counterexample, consider model M_2 from Figure 3.2A. Let $\overline{q}$ denote the state "opposite" to q, i.e. $\overline{q_1} = q_3$, $\overline{q_2} = q_4$ etc. Furthermore, let $S^i_{\mathbb{A}\text{gt}}$ denote the strategy of playing $\langle i, i, i, i\rangle$ in all states. Now, $S^i_{\mathbb{A}\text{gt}}$ is the only undominated strategy wrt $\overline{q_i}, \bigcirc\,\text{win}$ for $i = 1, ..., 4$, and $S^1_{\mathbb{A}\text{gt}}, ..., S^4_{\mathbb{A}\text{gt}}$ are exactly the strategies undominated wrt $q_0, \bigcirc\,\text{win}$. So, $M_2, q_i \models_{\text{ETSL}} \langle\langle\mathbb{A}\text{gt}\rangle\rangle\bigcirc\,\text{win}$ for every $i = 0, 1, ..., 4$, and therefore $M_2, q_0 \models_{\text{ETSL}} E_{\mathbb{A}\text{gt}}\langle\langle\mathbb{A}\text{gt}\rangle\rangle\bigcirc\,\text{win}$. On the other hand, there is no single strategy that succeeds for all $q_0, q_1, ..., q_4$.

Common knowledge: consider model M_3 from Figure 3.2B. Let $S_{\{a,b\}}$ be the strategy "play $\langle 1, 1\rangle$ everywhere", and $T_{\{a,b\}}$ be "play $\langle 2, 2\rangle$ everywhere". Note that $S_{\{a,b\}}$ is the only undominated strategy wrt $q, \bigcirc\,\text{win}$ for $q = q_0, q_1$, and $T_{\{a,b\}}$ is the only undominated strategy wrt $q, \bigcirc\,\text{win}$ for $q = q_2, q_3$. Thus, for every $q = q_0, ..., q_3$: $M_3, q \models_{\text{ETSL}} \langle\langle a, b\rangle\rangle\bigcirc\,\text{win}$, and hence $M_3, q_1 \models_{\text{ETSL}} C_{\{a,b\}}\langle\langle a, b\rangle\rangle\bigcirc\,\text{win}$. On the other hand, $M_3, q_1 \not\models_{\text{CSL}} \mathbb{C}_{\{a,b\}}\langle\langle a, b\rangle\rangle\bigcirc\,\text{win}$. ■

3.5 Conclusions

In this paper, the relationship between rational play and knowing how to play is investigated in a formal way. To this end, we dust off Epistemic Temporal Strategic Logic by van Otterloo and Jonker [135], and propose a simpler semantics expressed entirely in terms of concurrent epistemic game structures and their states; we prove that the new semantics is equivalent to the

original one for "vanilla" ETSL formulae. ETSL serves as a device for talking about the outcome of rational play (in the sense that agents are assumed to play only undominated strategies). To capture properties of the other kind ("knowing how to play"), we use the recent proposal of Constructive Strategic Logic [73, 75].

The main result of this paper states that, for finite models, *a rational player knows that he will succeed if, and only if, he knows how to succeed*. We also show that the relationship is much more limited for rational coalitions. That is, if rational agents have common knowledge about a winning strategy, then they have common knowledge that they will succeed – but the converse is not guaranteed any more. Moreover, it turns out that the relationship is *strictly reverse* for distributed knowledge: if a rational coalition has distributed knowledge that it will succeed, then it has distributed knowledge about a winning strategy – but not necessarily the other way around. Finally, for mutual knowledge, the relationship does not hold either way in general. This is a curious result, and one that may lead to interesting philosophical conclusions.

Part II

Plausible Behavior and Rational Play

Chapter 4

Agents, Beliefs, and Plausible Behavior in a Temporal Setting (joint work with Nils Bulling)

Abstract. Logics of knowledge and belief are often too static and inflexible to be used on real-world problems. In particular, they usually offer no concept for expressing that some course of events is *more likely to happen* than another. We address this problem and extend CTLK (computation tree logic with knowledge) with a notion of *plausibility*, which allows for practical and counterfactual reasoning. The new logic CTLKP (CTLK with plausibility) includes also a particular notion of belief. A plausibility update operator is added to this logic in order to change plausibility assumptions dynamically. Furthermore, we examine some important properties of these concepts. In particular, we show that, for a natural class of models, belief is a **KD45** modality. We also show that model checking CTLKP is **P**-complete and can be done in time linear with respect to the size of models and formulae.

`Keywords`: multi-agent systems, temporal logic, plausibility, beliefs

4.1 Introduction

Notions like *time*, *knowledge*, and *beliefs* are very important for analyzing the behavior of agents and multi-agent systems. In this paper, we extend modal logics of time and knowledge with a concept of *plausible behavior*: this notion is added to the language of CTLK [115], which is a straightforward combination of the branching-time temporal logic CTL [41, 38] and standard epistemic logic [57, 42].

In our approach, plausibility can be seen as a temporal property of behaviors. That is, some behaviors of the system can be assumed plausible

and others implausible, with the underlying idea that the latter should perhaps be ignored in practical reasoning about possible future courses of action. Moreover, behaviors can be formally understood as *temporal paths* in the Kripke structure modeling a multi-agent system. As a consequence, we obtain a language to reason about what can (or must) plausibly happen. We propose a particular notion of beliefs (inspired by [126, 46]), defined in terms of epistemic relations and plausibility. The main intuition is that beliefs are facts *that an agent would know if he assumed that only plausible things could happen*.

We believe that humans use such a concept of plausibility and "practical beliefs" quite often in their everyday reasoning. Restricting one's reasoning to plausible possibilities is essential to make the reasoning feasible, as the space of *all* possibilities is exceedingly large in real life. We investigate some important properties of plausibility, knowledge, and belief in this new framework. In particular, we show that knowledge is an **S5** modality, and that beliefs satisfy axioms **K45** in general, and **KD45** for the class of *plausibly serial models*. Finally, we show that the relationship between knowledge and belief for plausibly serial models is natural and reflects the initial intuition well. We also show how plausibility assumptions can be specified in the object language via a *plausibility update operator*, and we study properties of such updates. Finally, we show that model checking of the new logic is no more complex than model checking CTL and CTLK.

Our ultimate goal is to come up with a logic that allows the study of strategies, time, knowledge, and plausible/rational behavior under both perfect and imperfect information. As combining all these dimensions is highly nontrivial (cf. [83, 73]) it seems reasonable to split this task. While this paper deals with knowledge, plausibility, and belief, the companion paper [76] proposes a general framework for multi-agent systems that regard game-theoretical rationality criteria like Nash equilibrium, Pareto optimality, etc. The latter approach is based on the more powerful logic ATL [8].

The paper is structured as follows. Firstly, we briefly present branching-time logic with knowledge, CTLK. In Section 3 we present our approach to plausibility and formally define CTLK with plausibility. We also show how temporal formulae can be used to describe plausible paths, and we compare our logic with existing related work. In Section 4.4, properties of knowledge, belief, and plausibility are explored. Finally, we present verification complexity results for CTLKP in Section 4.5.

4.2 Branching Time and Knowledge

In this paper we develop a framework for agents' beliefs about how the world can (or must) evolve. Thus, we need a notion of time and change, plus a notion of what the agents are supposed to know in particular situations. CTLK [115] is a straightforward combination of the computation tree logic CTL [41, 38] and standard epistemic logic [57, 42].

CTL includes operators for temporal properties of systems: i.e., path quantifier E ("there is a path"), together with temporal operators: $\bigcirc$ ("in the next

state"), $\Box$ ("always from now on") and $\mathcal{U}$ ("until").[1] Every occurrence of a temporal operator is preceded by exactly one path quantifier in CTL (this variant of the language is sometimes called "vanilla" CTL). Epistemic logic uses operators for representing agents' knowledge: $K_a\varphi$ is read as "agent a knows that φ".

Let Π be a set of atomic propositions with a typical element p, and $\mathbb{A}\mathrm{gt} = \{1, ..., k\}$ be a set of agents with a typical element a. The language of CTLK consists of formulae φ, given as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{E}\gamma \mid K_a\varphi$$
$$\gamma ::= \bigcirc \varphi \mid \Box\varphi \mid \varphi \mathcal{U} \varphi.$$

We will sometimes refer to formulae φ as ("vanilla") *state* formulae and to formulae γ as ("vanilla") *path* formulae.

The semantics of CTLK is based on Kripke models $\mathcal{M} = \langle St, R, \sim_1, ..., \sim_k, \pi\rangle$, which include a nonempty set of states St, a state transition relation $R \subseteq St \times St$, epistemic indistinguishability relations $\sim_a \subseteq St \times St$ (one per agent), and a valuation of propositions $\pi : \Pi \to 2^{(}St)$. We assume that relation R is serial and that all $\sim_a$ are equivalence relations. A *path* λ in $\mathcal{M}$ refers to a possible behavior (or computation) of system $\mathcal{M}$, and can be represented as an infinite sequence of states that follow relation R, that is, a sequence $q_0q_1q_2...$ such that q_iRq_{i+1} for every $i = 0, 1, 2, ...$ We denote the ith state in λ by $\lambda[i]$. The set of all paths in $\mathcal{M}$ is denoted by $\Lambda_\mathcal{M}$ (if the model is clear from context, $\mathcal{M}$ will be omitted). A q-*path* is a path that starts from q, i.e., $\lambda[0] = q$. A q-*subpath* is a sequence of states, starting from q, which is a subpath of some path in the model, i.e. a sequence $q[0]q[1]...$ such that $q = q[0]$ and there are $q^0, ..., q^i$ such that $q^0...q^iq[0]q[1]... \in \Lambda_\mathcal{M}$.[2] The semantics of CTLK is defined as follows:

$\mathcal{M}, q \models p$ iff $q \in \pi(p)$;

$\mathcal{M}, q \models \neg\varphi$ iff $\mathcal{M}, q \not\models \varphi$;

$\mathcal{M}, q \models \varphi \wedge \psi$ iff $\mathcal{M}, q \models \varphi$ and $\mathcal{M}, q \models \psi$;

$\mathcal{M}, q \models \mathsf{E}\bigcirc\varphi$ iff there is a q-path λ such that $\mathcal{M}, \lambda[1] \models \varphi$;

$\mathcal{M}, q \models \mathsf{E}\Box\varphi$ iff there is a q-path λ such that $\mathcal{M}, \lambda[i] \models \varphi$ for every $i \geq 0$;

$\mathcal{M}, q \models \mathsf{E}\varphi\mathcal{U}\psi$ iff there is a q-path λ and $i \geq 0$ such that $\mathcal{M}, \lambda[i] \models \psi$, and $\mathcal{M}, \lambda[j] \models \varphi$ for every $0 \leq j < i$;

$\mathcal{M}, q \models K_a\varphi$ iff $\mathcal{M}, q \models \varphi$ for every q' such that $q \sim_a q'$.

4.3 Extending Time and Knowledge with Plausibility and Beliefs

In this section we discuss the central concept of this paper, i.e. the concept of plausibility. First, we outline the idea informally. Then, we extend CTLK

[1] Additional operators A ("for every path") and $\Diamond$ ("sometime in the future") are defined in the usual way.

[2] For CTLK models, λ is a q-subpath iff it is a q-path. It will not always be so when plausible paths are introduced.

with the notion of plausibility by adding *plausible path operators* $\mathbf{Pl}_a$ and *physical path operator* $\mathbf{Ph}$ to the logic. Formula $\mathbf{Pl}_a\varphi$ has the intended meaning: *according to agent a, it is plausible that φ holds*; formula $\mathbf{Ph}\,\varphi$ reads as: *φ holds in all "physically" possible scenarios* (i.e., even in implausible ones). The plausible path operator restricts statements only to those paths which are defined to be "sensible", whereas the physical path operator generates statements about all paths that may theoretically occur. Furthermore, we define beliefs on top of plausibility and knowledge, as the facts *that an agent would know if he assumed that only plausible things could happen*. Finally, we discuss related work [46, 47, 126, 107, 93], and compare it with our approach.

4.3.1 The Concept of Plausibility

It is well known how knowledge (or beliefs) can be modeled with Kripke structures. However, it is not so obvious how we can capture knowledge *and* beliefs in a sensible way in *one* framework. Clearly, there should be a connection between these two notions. Our approach is to use the notion of plausibility for this purpose. Plausibility can serve as a primitive concept that helps to define the semantics of beliefs, in a similar way as indistinguishability of states (represented by relation $\sim_a$) is the semantic concept that underlies knowledge. In this sense, our work follows [46, 126]: essentially, beliefs are what an agent would know if he took only plausible options into account. In our approach, however, plausibility is explicitly seen as a *temporal property*. That is, we do not consider states (or possible worlds) to be more plausible than others but rather define some behaviors to be plausible, and others implausible. Moreover, behaviors can be formally understood as temporal paths in the Kripke structure modeling a multi-agent system.

An actual notion of plausibility (that is, a particular set of plausible paths) can emerge in many different ways. It may result from observations and learning; an agent can learn from its observations and see specific patterns of events as plausible ("a lot of people wear black shoes if they wear a suit"). Knowledge exchange is another possibility (e.g., an agent a can tell agent b that "player c always bluffs when he is smiling"). Game theory, with its rationality criteria (undominated strategies, maxmin, Nash equilibrium etc.) is another viable source of plausibility assumptions. Last but not least, folk knowledge can be used to establish plausibility-related classifications of behavior ("players normally want to win a game", "people want to live").

In any case, restricting the reasoning to plausible possibilities can be essential if we want to make the reasoning feasible, as the space of *all* possibilities (we call them "physical" possibilities in the rest of the paper) is exceedingly large in real life. Of course, this does not exclude a more extensive analysis in special cases, e.g. when our plausibility assumptions do not seem accurate any more, or when the cost of inaccurate assumptions can be too high (as in the case of high-budget business decisions). But even in these cases, we usually do not get rid of plausibility assumptions completely – we only revise them to make them more cautious.[3]

[3]That is, when planning to open an industrial plant in the UK, we will probably consider the possibility of our main contractor taking her life, but we will still *not* take into account the possibilities of: an invasion of UFO, England being destroyed by a meteorite, Fidel Castro

To formalize this idea, we extend models of CTLK with *sets of plausible paths* and add plausibility operators $\mathbf{Pl}_a$, physical paths operator $\mathbf{Ph}$, and belief operators B_a to the language of CTLK. Now, it is possible to make statements that refer to plausible paths only, as well as statements that regard all paths that may occur in the system.

4.3.2 CTLK with Plausibility

In this section, we extend the logic of CTLK with plausibility; we call the resulting logic CTLKP. Formally, the language of CTLKP is defined as:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{E}\gamma \mid \mathbf{Pl}_a\varphi \mid \mathbf{Ph}\,\varphi \mid K_a\varphi \mid B_a\varphi$$
$$\gamma ::= \bigcirc\varphi \mid \Box\varphi \mid \varphi\,\mathcal{U}\,\varphi.$$

For instance, we may claim it is plausible to assume that a shop is closed after the opening hours, though the manager may be physically able to open it at any time: $\mathbf{Pl}_a\mathsf{A}\Box(\mathsf{late} \rightarrow \neg\mathsf{open}) \wedge \mathbf{Ph}\,\mathsf{E}\Diamond(\mathsf{late} \wedge \mathsf{open})$.

The semantics of CTLKP extends that of CTLK as follows. Firstly, we augment the models with *sets of plausible paths*. A *model with plausibility* is given as

$$\mathcal{M} = \langle St, R, \sim_1, ..., \sim_k, \Upsilon_1, ..., \Upsilon_k, \pi\rangle,$$

where $\langle St, R, \sim_1, ..., \sim_k, \pi\rangle$ is a CTLK model, and $\Upsilon_a \subseteq \Lambda_{\mathcal{M}}$ is the set of paths in $\mathcal{M}$ that are plausible according to agent a. If we want to make it clear that Υ_a is taken from model $\mathcal{M}$, we will write $\Upsilon_a^{\mathcal{M}}$. It seems worth emphasizing that this notion of plausibility is *subjective* and *holistic*. It is subjective because Υ_a represents *agent a's subjective view on what is plausible* – and indeed, different agents may have different ideas on plausibility (i.e., Υ_a may differ from Υ_b). It is holistic because Υ_a represents agent a's idea of the plausible behavior *of the whole system* (including the behavior of other agents).

Remark 13 *In our models, plausibility is also* global*, i.e., plausibility sets do not depend on the state of the system. Investigating systems, in which plausibility is relativized with respect to states (like in [46]), might be an interesting avenue of future work. However, such an approach – while obviously more flexible – allows for potentially counterintuitive system descriptions. For example, it might be the case that path λ is plausible in $q = \lambda[0]$, but the set of plausible paths in $q' = \lambda[1]$ is empty. That is, by following plausible path λ we are bound to get to an implausible situation. But then, does it make sense to consider λ as plausible?*

Secondly, we use a non-standard satisfaction relation $\models_P$, which we call *plausible satisfaction*. Let $\mathcal{M}$ be a CTLKP model and $P \subseteq \Lambda_{\mathcal{M}}$ be an arbitrary subset of paths in $\mathcal{M}$ (not necessarily any $\Upsilon_a^{\mathcal{M}}$). $\models_P$ restricts the evaluation of temporal formulae to the paths given in P only. The "absolute" satisfaction relation $\models$ is defined as $\models_{\Lambda_{\mathcal{M}}}$.

Let $\partial(P)$ be the set of all states that lie on at least one path in P, i.e. $\partial(P) = \{q \in St \mid \exists\lambda \in P \exists i\,(\lambda[i] = q)\}$. Now, the semantics of CTLKP can be given through the following clauses:

becoming the British Prime Minister etc. Note that this is fundamentally different from using a probabilistic model in which all these unlikely scenarios are assigned very low probabilities: in that case, they also have a very small influence on our final decision, but we must process the *whole* space of physical possibilities to evaluate the options.

$\mathcal{M}, q \models_P p$ iff $q \in \pi(p)$;

$\mathcal{M}, q \models_P \neg\varphi$ iff $\mathcal{M}, q \not\models_P \varphi$;

$\mathcal{M}, q \models_P \varphi \wedge \psi$ iff $\mathcal{M}, q \models_P \varphi$ and $\mathcal{M}, q \models_P \psi$;

$\mathcal{M}, q \models_P \mathsf{E}\bigcirc\varphi$ iff there is a q-subpath $\lambda \in P$ such that $\mathcal{M}, \lambda[1] \models_P \varphi$;

$\mathcal{M}, q \models_P \mathsf{E}\Box\varphi$ iff there is a q-subpath $\lambda \in P$ such that $\mathcal{M}, \lambda[i] \models_P \varphi$ for every $i \geq 0$;

$\mathcal{M}, q \models_P \mathsf{E}\varphi\mathcal{U}\psi$ iff there is a q-subpath $\lambda \in P$ and $i \geq 0$ such that $\mathcal{M}, \lambda[i] \models_P \psi$, and $\mathcal{M}, \lambda[j] \models_P \varphi$ for every $0 \leq j < i$;

$\mathcal{M}, q \models_P \mathbf{Pl}_a\varphi$ iff $\mathcal{M}, q \models_{\Upsilon_a} \varphi$;

$\mathcal{M}, q \models_P \mathbf{Ph}\,\varphi$ iff $\mathcal{M}, q \models \varphi$;

$\mathcal{M}, q \models_P K_a\varphi$ iff $\mathcal{M}, q \models \varphi$ for every q' such that $q \sim_a q'$;

$\mathcal{M}, q \models_P B_a\varphi$ iff for all $q' \in \partial(\Upsilon_a)$ with $q \sim_a q'$, we have that $\mathcal{M}, q' \models_{\Upsilon_a} \varphi$.

One of the main reasons for using the concept of plausibility is that we want to define agents' *beliefs* out of more primitive concepts – in our case, these are plausibility and indistinguishability – in a way analogous to [126, 46]. If an agent *knows* that φ, he must be "sure" about it. However, *beliefs* of an agent are not necessarily about reliable facts. Still, they should make sense to the agent; if he believes that φ, then the formula should at least hold in all futures that he envisages as plausible. Thus, beliefs of an agent may be seen as *things known to him if he disregards all non-plausible possibilities*.

We say that φ is $\mathcal{M}$*-true* ($\mathcal{M} \models \varphi$) if $\mathcal{M}, q \models \varphi$ for all $q \in St_{\mathcal{M}}$. φ is *valid* ($\models \varphi$) if $\mathcal{M} \models \varphi$ for all models $\mathcal{M}$. φ is $\mathcal{M}$*-strongly true* ($\mathcal{M} \Vdash \varphi$) if $\mathcal{M}, q \models_P \varphi$ for all $q \in St_{\mathcal{M}}$ and all $P \subseteq \Lambda_{\mathcal{M}}$. φ is *strongly valid* ($\Vdash \varphi$) if $\mathcal{M} \Vdash \varphi$ for all models $\mathcal{M}$.

Proposition 33 *Strong truth and strong validity imply truth and validity, respectively. The reverse does not hold.*

Ultimately, we are going to be interested in normal (not strong) validity, as parameterizing the satisfaction relation with a set P is just a technical device for propagating sets of plausible paths Υ_a into the semantics of nested formulae. The importance of strong validity, however, lies in the fact that $\Vdash \varphi \leftrightarrow \psi$ makes φ and ψ completely interchangeable, while the same is not true for normal validity.

Proposition 34 *Let $\Phi[\varphi/\psi]$ denote formula Φ in which every occurrence of ψ was replaced by φ. Also, let $\Vdash \varphi \leftrightarrow \psi$. Then for all M, q, P: $M, q \models_P \Phi$ iff $M, q \models_P \Phi[\varphi/\psi]$ (in particular, $M, q \models \Phi$ iff $M, q \models \Phi[\varphi/\psi]$).*

Note that $\models \varphi \leftrightarrow \psi$ does not even imply that $M, q \models \Phi$ iff $M, q \models \Phi[\varphi/\psi]$.

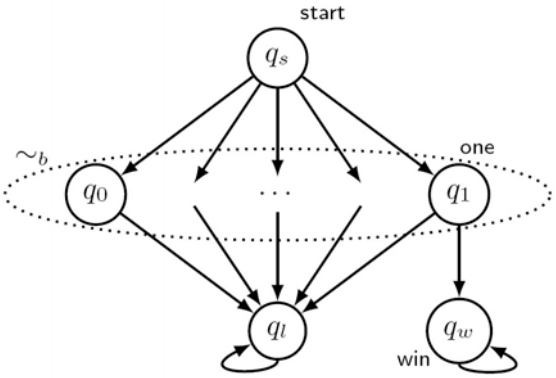


Figure 4.1: Guessing Robots game

Example 15 (Guessing Robots) *Consider a simple game with two agents a and b, shown in Figure 4.1. First, a chooses a real number $r \in [0,1]$ (without revealing the number to b); then, b chooses a real number $r' \in [0,1]$. The agents win the game (and collect EUR $1,000,000$) if both chose 1, otherwise they lose. Formally, we model the game with a CTLKP model $\mathcal{M}$, in which the set of states St includes $q[s]$ for the initial situation, states $q[r]$, $r \in [0,1]$, for the situations after a has chosen number r, and "final" states $q[w], q[l]$ for the winning and the losing situation, respectively. The transition relation is as follows: $q[s]Rq[r]$ and $q[r]Rq[l]$ for all $r \in [0,1]$; $q[1]Rq[w]$, $q[w]Rq[w]$, and $q[l]Rq[l]$. Moreover, $\pi(\mathsf{one}) = \{q[1]\}$ and $\pi(\mathsf{win}) = \{q[w]\}$. Player a has perfect information in the game (i.e., $q \sim_a q'$ iff $q = q'$), but player b does not distinguish between states $q[r]$ (i.e., $q[r] \sim_b q[r']$ for all $r, r' \in [0,1]$). Obviously, the only sensible thing to do for both agents is to choose 1 (using game-theoretical vocabulary, these strategies are* strongly dominant *for the respective players). Thus, there is only one* plausible *course of events if we assume that our players are rational, and hence $\Upsilon_a = \Upsilon_b = \{q[s]q[1]q[w]q[w]\ldots\}$.*

Note that, in principle, the outcome of the game is uncertain: $\mathcal{M}, q[s] \models \neg\mathsf{A}\Diamond\mathsf{win} \wedge \neg\mathsf{A}\Box\neg\mathsf{win}$. However, assuming rationality of the players makes it only plausible that the game must end up with a win: $\mathcal{M}, q[s] \models \mathbf{Pl}_a\,\mathsf{A}\Diamond\mathsf{win} \wedge \mathbf{Pl}_b\,\mathsf{A}\Diamond\mathsf{win}$, and the agents believe that this will be the case: $\mathcal{M}, q[s] \models B_a\mathsf{A}\Diamond\mathsf{win} \wedge B_b\mathsf{A}\Diamond\mathsf{win}$. Note also that, in any of the states $q[r]$, agent b believes that a (being rational) has played 1: $\mathcal{M}, q[r] \models B_b\mathsf{one}$ for all $r \in [0,1]$.

4.3.3 Defining Plausible Paths with Formulae

So far, we have assumed that sets of plausible paths are somehow given in models. In this section we present a dynamic approach where an actual notion of plausibility can be specified in the object language. Note that we want to specify (usually infinite) sets of infinite paths, and we need a finite representation of these structures. One logical solution is given by using path formulae γ. These formulae describe properties of paths; therefore, a specific formula can be used to characterize a set of paths. For instance, think about a country in Africa where it has never snowed. Then, plausible paths might be defined as ones in which it never snows, i.e., all paths that satisfy $\Box\neg\mathsf{snows}$. Formally, let γ be a CTLK path formula. We define $|\gamma|_{\mathcal{M}}$ to

be the set of paths that satisfy γ in model $\mathcal{M}$:

$$\begin{aligned} |\bigcirc\varphi|_{\mathcal{M}} &= \{\lambda \mid \mathcal{M}, \lambda[1] \models \varphi\} \\ |\Box\varphi|_{\mathcal{M}} &= \{\lambda \mid \forall i\, (\mathcal{M}, \lambda[i] \models \varphi)\} \\ |\varphi_1 \mathcal{U} \varphi_2|_{\mathcal{M}} &= \{\lambda \mid \exists i \big(\mathcal{M}, \lambda[i] \models \varphi_2 \wedge \\ &\qquad \forall j(0 \leq j < i \Rightarrow \mathcal{M}, \lambda[j] \models \varphi_1)\big)\}. \end{aligned}$$

Moreover, we define the *plausible paths model update* as follows. Let $\mathcal{M} = \langle St, R, \sim_1, ..., \sim_k, \Upsilon_1, ..., \Upsilon_k, \pi\rangle$ be a CTLKP model, and let $P \subseteq \Lambda_{\mathcal{M}}$ be a set of paths. Then $\mathcal{M}^{a,P} = \langle St, R, \sim_1, ..., \sim_k, \Upsilon_1, ..., \Upsilon_{a-1}, P, \Upsilon_{a+1}, ..., \Upsilon_k, \pi\rangle$ denotes model $\mathcal{M}$ with a's set of plausible paths reset to P.

Now we can extend the language of CTLKP with formulae (**set-pl**$_a$ $\gamma)\varphi$ with the intuitive reading: "suppose that γ exactly characterizes the set of plausible paths, then φ holds", and formal semantics given below:

$$\mathcal{M}, q \models_P (\textbf{set-pl}_a\ \gamma)\varphi \text{ iff } \mathcal{M}^{a,|\gamma|_{\mathcal{M}}}, q \models_P \varphi.$$

We observe that this update scheme is similar to the one proposed in [85].

4.3.4 Comparison to Related Work

Several modal notions of plausibility were already discussed in the existing literature [46, 47, 126, 107, 93]. In these papers, like in ours, plausibility is used as a primitive semantic concept that helps to define beliefs on top of agents' knowledge. A similar idea was introduced by Moses and Shoham in [107]. Their work preceded both [46, 47] and [126] – and although Moses and Shoham do not explicitly mention the term "plausibility", it seems appropriate to summarize their idea first.

Moses and Shoham: Beliefs as Conditional Knowledge

In [107], beliefs are relativized with respect to a formula α (which can be seen as a plausibility assumption expressed in the object language). More precisely, worlds that satisfy α can be considered as plausible. This concept is expressed via symbols $B_i^\alpha\varphi$; the index $i \in \{1, 2, 3\}$ is used to distinguish between three different implementations of beliefs. The first version is given by $B_1^\alpha\varphi \equiv K(\alpha \rightarrow \varphi)$.[4] A drawback of this version is that if α is false, then everything will be believed with respect to α. The second version overcomes this problem: $B_2^\alpha\varphi \equiv K(\alpha \rightarrow \varphi) \wedge (K\neg\alpha \rightarrow K\varphi)$; now φ is only believed if it is known that φ follows from assumption α, and φ must be known if assumption α is known to be false. Finally, $B_3^\alpha\varphi \equiv K(\alpha \rightarrow \varphi) \wedge \neg K\neg\alpha$: if the assumption α is known to be false, nothing should be believed with respect to α. The strength of these different notions is given as follows: $B_3^\alpha\varphi$ implies $B_2^\alpha\varphi$, and $B_2^\alpha\varphi$ implies $B_1^\alpha\varphi$. In this approach, belief is strongly connected to knowledge in the sense that belief is knowledge with respect to a given assumption.

[4] Unlike in most approaches, K is interpreted over *all* worlds and not only over the indistinguishable worlds.

Friedman and Halpern: Plausibility Spaces

The work of Friedman and Halpern [46] extends the concepts of knowledge and belief with an *explicit* notion of plausibility; i.e., some worlds are more plausible for an agent than others. To implement this idea, Kripke models are extended with function P which assigns a *plausibility space* $P(q,a) = (\Omega_{(q,a)}, \preceq_{(q,a)})$ to every state, or more generally every possible world q, and agent a. The plausibility space is just a partially ordered subset of states/worlds; that is, $\Omega(q,a) \subseteq St$, and $\preceq_{(q,a)} \subseteq St \times St$ is a reflexive and transitive relation. Let $S, T \subseteq \Omega_{(q,a)}$ be finite subsets of states; now, T is defined to be *plausible given S with respect to* $P(q,a)$, denoted by $S \rightarrow_{P(q,a)} T$, iff all minimal points/states in S (with respect to $\preceq_{(q,a)}$) are also in T.[5] Friedman and Halpern's view to modal plausibility is closely related to probability and, more generally, plausibility measures. Logics of plausibility can be seen as a *qualitative* description of agents preferences/knowledge; logics of probability [43, 91], on the other hand, offer a *quantitative* description.

The logic from [46] is defined by the following grammar: $\varphi ::= p \mid \varphi \wedge \varphi \mid \neg\varphi \mid K_a\varphi \mid \varphi \rightarrow_a \varphi$, where the semantics of all operators except $\rightarrow_a$ is given as usual, and formulae $\varphi \rightarrow_a \psi$ have the meaning that ψ is true in the most plausible worlds in which φ holds. Formally, the semantics for $\rightarrow_a$ is given as: $\mathcal{M}, q \models \varphi \rightarrow_a \psi$ iff $S^{\varphi}_{P(q,a)} \rightarrow_{P_{(q,a)}} S^{\psi}_{P(q,a)}$, where $S^{\varphi}_{(q,a)} = \{q' \in \Omega_{(q,a)} \mid \mathcal{M}, q' \models \varphi\}$ are the states in $\Omega_{(q,a)}$ that satisfy φ. The idea of defining beliefs is given by the assumption that an agent *believes* in something if he *knows that it is true in the most plausible worlds of* $\Omega_{(q,a)}$; formally, this can be stated as $B_a\varphi \equiv K_a(\top \rightarrow_a \varphi)$.

Friedman and Halpern have shown that the **KD45** axioms are valid for operator B_a if plausibility spaces satisfy *consistency* (for all states $q \in St$ it holds that $\Omega_{(q,a)} \subseteq \{ q' \in St \mid q \sim_a q' \}$) and *normality* (for all states $q \in St$ it holds that $\Omega_{(q,a)} \neq \emptyset$).[6] A temporal extension of the language (mentioned briefly in [46], and discussed in more detail in [47]) uses the interpreted systems approach [58, 42]. A *system* $\mathcal{R}$ is given by *runs*, where a run $r : \mathbb{N} \rightarrow St$ is a function from time moments (modeled by $\mathbb{N}$) to global states, and a *time point* (r,i) is given by a time point $i \in \mathbb{N}$ and a run r. A global state is a combination of local states, one per agent. An *interpreted system* $\mathcal{M} = (\mathcal{R}, \pi)$ is given by a system $\mathcal{R}$ and a valuation of propositions π. Epistemic relations are defined over time points, i.e., $(r', m') \sim_a (r, m)$ iff agent a's local states $r'_a(m')$ and $r_a(m)$ of (r', m') and (r, m) are equal. Formulae are interpreted in a straightforward way with respect to interpreted systems, e.g. $\mathcal{M}, r, m \models K_a\varphi$ iff $\mathcal{M}, r', m' \models \varphi$ for all $(r', m') \sim_a (r, m)$. Now, these are time points that play the role of possible worlds; consequently, plausibility spaces $\mathcal{P}_{(r,m,a)}$ are assigned to each point (r, m) and agent a.

Su et al.: KBC Logic

Su et al. [126] have developed a multi-modal, computationally grounded logic with modalities K, B, and C (knowledge, belief, and certainty). The

[5] When there are infinite chains $\ldots \preceq q_3 \preceq q_2 \preceq_a q_1$, the definition is much more sophisticated. An interested reader is referred to [46] for more details.

[6] Note that this "normality" is essentially seriality of states wrt plausibility spaces.

computational model consists of (global) states $q = (q^{vis}, q^{inv}, q^{per}, St^{pls})$ where the environment is divided into a visible (q^{vis}) and an invisible part (q^{inv}), and q^{per} captures the agent's perception of the visible part of the environment. External sources may provide the agent with information about the invisible part of a state, which results in a set of states St^{pls} that are plausible for the agent.

Given a global state q, we additionally define $Vis(q) = q^{vis}$, $Inv(q) = q^{inv}$, $Per(q) = q^{per}$, and $Pls(q) = St^{pls}$. The semantics is given by an extension of interpreted systems [58, 42], here, it is called *interpreted KBC systems*. KBC formulae are defined as $\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K\varphi \mid B\varphi \mid C\varphi$. The epistemic relation $\sim_{vis}$ is captured in the following way: $(r,i) \sim_{vis} (r',i')$ iff $Vis(r(i)) = Vis(r'(i'))$. The semantic clauses for belief and certainty are given below.

$\mathcal{M}, r, i \models B\varphi$ iff $\mathcal{M}, r', i' \models \varphi$ for all (r', i') with $Vis(r'(i')) = Per(r(i))$ and $Inv(r'(i')) \in Pls(r(i))$

$\mathcal{M}, r, i \models C\varphi$ iff $\mathcal{M}, r', i' \models \varphi$ for all $(r'(i'))$ with $Vis(r'(i')) = Per(r(i))$

Thus, an agent believes φ if, and only if, φ is true in all states *which look like what he sees now and seem plausible in the current state*. Certainty is stronger: if an agent is certain about φ, the formula must hold in all states with a visible part equal to the current perception, regardless of whether the invisible part is plausible or not.

The logic does not include temporal formulae, although it might be extended with temporal operators, as time is already present in KBC models.

What Are the Differences to Our Logic?

In our approach, plausibility is explicitly seen as a temporal property, i.e., it is a property of temporal paths rather than states. In the object language, this is reflected by the fact that plausibility assumptions are specified through path formulae. In contrast, the approach of [107] and [126] is static: not only the logics do not include operators for talking about time and/or change, but these are *states* that are assumed plausible or not in their semantics.

The differences to [46, 47] are more subtle. Firstly, the framework of Friedman and Halpern is static in the sense that plausibility is taken as a property of (abstract) possible worlds. This formulation is flexible enough to allow for incorporating time; still, in our approach, time is *inherent* to plausibility rather than incidental.

Secondly, our framework is more computationally oriented. The implementation of temporal plausibility in [46, 47] is based on the interpreted systems approach with *time points* (r, m) being subject to plausibility. As runs are included in time points, they can also be defined plausible or implausible.[7] However, it also means that time points serve the role of possible worlds in the basic formulation, which yields Kripke structures with *uncountable* possible world spaces in all but the most trivial cases.

Thirdly, [46, 47] build on *linear* time: a run (more precisely, a time moment (r, m)) is fixed when a formula is interpreted. In contrast, we use branch-

[7] Friedman and Halpern even briefly mention how plausibility of runs can be embedded in their framework.

ing time with explicit quantification over temporal paths.[8] We believe that branching time is more suitable for non-deterministic domains (cf. e.g. [41]), of which multi-agent systems are a prime example. Note that branching time makes our notion of belief different from Friedman and Halpern's. Most notably, property $K\varphi \rightarrow B\varphi$ is valid in their approach, but not in ours: an agent may know that some course of events is in principle possible, without believing that it can *really* become the case (see Section 4.4.2). As Proposition 42 suggests, such a subtle distinction between knowledge and beliefs is possible in our approach because branching time logics allow for existential quantification over runs.

Fourthly, while Friedman and Halpern's models are very flexible, they also enable system descriptions that may seem counterintuitive. Suppose that (r, m) is plausible in itself (formally: (r, m) is minimal wrt $\preceq_{(r,m,a)}$), but $(r, m{+}1)$ is not plausible in $(r, m{+}1)$. This means that following the plausible path makes it implausible (cf. Remark 13), which is even stranger in the case of linear time. Combining the argument with computational aspects, we suggest that our approach can be more natural and straightforward for many applications.

Last but not least, our logic provides a mechanism for specifying (and updating) sets of plausible paths in the object language. Thus, plausibility sets can be specified in a succinct way, which is another feature that makes our framework computation-friendly. The model checking results from Section 4.5 are especially encouraging in this light.

4.4 Plausibility, Knowledge, and Beliefs in CTLKP

In this section we study some relevant properties of plausibility, knowledge, and beliefs; in particular, axioms **KDT45** are examined. But first, we identify two important subclasses of models with plausibility.

A CTLKP model is *plausibly serial* (or *p-serial*) for agent a if every state of the system is part of a plausible path according to a, i.e. $\partial(\Upsilon_a) = St$. As we will see further, a weaker requirement is sometimes sufficient. We call a model *weakly p-serial* if every state has at least one indistinguishable counterpart which lies on a plausible path, i.e. for each $q \in St$ there is a $q' \in St$ such that $q \sim_a q'$ and $q' \in \partial(\Upsilon_a)$. Obviously, p-seriality implies weak p-seriality. We get the following characterization of both model classes.

Proposition 35 *M is plausibly serial for agent a iff formula $\mathbf{Pl}_a \mathsf{E}\bigcirc \top$ is valid in M. M is weakly p-serial for agent a iff $\neg K_a \mathbf{Pl}_a \mathsf{A}\bigcirc \bot$ is valid in M.*

4.4.1 Axiomatic Properties

Theorem 15 *Axioms **K**, **D**, **4**, and **5** for knowledge are strongly valid, and axiom **T** is valid. That is, modalities K_a form system **S5** in the sense of normal*

[8] To be more precise, time in [46] does implicitly branch at epistemic states. This is because $(r, m) \sim_a (r', m')$ iff a's local state corresponding to both time points is the same ($r_a(m) = r'_a(m')$). In consequence, the semantics of $K_a\varphi$ can be read as "*for every run*, and every moment on this run that yields the same local state as now, φ holds".

*validity, and **KD45** in the sense of strong validity.*

We do not include proofs here due to lack of space. The interested reader is referred to [24], where detailed proofs are given.

Proposition 36 *Axioms **K**, **4**, and **5** for beliefs are strongly valid. That is, we have:*
$\models (B_a\varphi \wedge B_a(\varphi \to \psi)) \to B_a\psi$, $\models (B_a\varphi \to B_aB_a\varphi)$, *and* $\models (\neg B_a\varphi \to B_a\neg B_a\varphi)$.

The next proposition concerns the "consistency" axiom **D**: $B_a\varphi \to \neg B_a\neg\varphi$. It is easy to see that the axiom is not valid in general: as we have no restrictions on plausibility sets Υ_a, it may be as well that $\Upsilon_a = \emptyset$. In that case we have $B_a\varphi \wedge B_a\neg\varphi$ for all formulae φ, because the set of states to be considered becomes empty. However, it turns out that **D** *is* valid for a very natural class of models.

Proposition 37 *Axiom **D** for beliefs is not valid in the class of all CTLKP models. However, it is strongly valid in the class of weak p-serial models (and therefore also in the class of p-serial models).*

Moreover, as one may expect, beliefs do not have to be always true.

Proposition 38 *Axiom **T** for beliefs is not valid; i.e.,* $\not\models (B_a\varphi \to \varphi)$. *The axiom is not even valid in the class of p-serial models.*

Theorem 16 *Belief modalities B_a form system **K45** in the class of all models, and **KD45** in the class of weakly plausibly serial models (in the sense of both normal and strong validity). Axiom **T** is not even valid for p-serial models.*

4.4.2 Plausibility, Knowledge, and Beliefs

First, we investigate the relationship between knowledge and plausibility/ physicality operators. Then, we look at the interaction between knowledge and beliefs.

Proposition 39 *Let φ be a CTLKP formula, and $\mathcal{M}$ be a CTLKP model. We have the following strong validities:*

1. $\models \mathbf{Pl}_a K_a\varphi \leftrightarrow K_a\varphi$
2. $\models \mathbf{Ph}\, K_a\varphi \leftrightarrow K_a\mathbf{Ph}\,\varphi$ *and* $\models K_a\mathbf{Ph}\,\varphi \leftrightarrow K_a\varphi$

We now want to examine the relationship between knowledge and belief. For instance, if agent a believes in something, he knows that he believes it. Or, if he knows a fact, he also believes that he knows it. On the other hand, for instance, an agent does not necessarily believe in all the things he knows. For example, we may know that an invasion from another galaxy is in principle possible ($K_a\mathsf{E}\Diamond$ invasion), but if we do not take this possibility as plausible ($\neg\mathbf{Pl}_a\mathsf{E}\Diamond$invasion), then we reject the corresponding belief in consequence ($\neg B_a\mathsf{E}\Diamond$invasion). Note that this property reflects the strong connection between belief and plausibility in our framework.

Proposition 40 *The following formulae are strongly valid:*
(i) $B_a\varphi \to K_aB_a\varphi$, *(ii)* $K_aB_a\varphi \to B_a\varphi$,
(iii) $K_a\varphi \to B_aK_a\varphi$.
The following formulae are not *valid:*
(iv) $B_a\varphi \to B_aK_a\varphi$, *(v)* $K_a\varphi \to B_a\varphi$

The last invalidity is especially important: it is *not* the case that knowing something implies believing in it. This emphasizes that we study a specific concept of beliefs here. Note that its specific is not due to the plausibility-based definition of beliefs. The reason lies rather in the fact that we investigate knowledge, beliefs and plausibility in a temporal framework, as Proposition 41 shows.

Proposition 41 *Let φ be a CTLKP formula that does not include any temporal operators. Then $K_a\varphi \to B_a\varphi$ is strongly valid, and in the class of p-serial models we have even that $\models K_a\varphi \leftrightarrow B_a\varphi$.*

Moreover, it is important that we use branching time with explicit quantification over paths; this observation is formalized in Proposition 42.

Definition 7 *We define the* universal sublanguage *of CTLK in a way similar to [132]:*

$$\varphi_u ::= p \mid \neg p \mid \varphi_u \wedge \varphi_u \mid \varphi_u \vee \varphi_u \mid \mathsf{A}\gamma_u \mid K_a\varphi_u,$$
$$\gamma_u ::= \bigcirc \varphi_u \mid \Box\varphi_u \mid \varphi_u \mathcal{U} \varphi_u.$$

We call such φ_u universal formulae, and γ_u universal path formulae.

Proposition 42 *Let φ_u be a universal CTLK formula. Then $\models K_a\varphi_u \to B_a\varphi_u$.*

The following two theorems characterize the relationship between knowledge and beliefs: first for the class of p-serial models, and then, finally, for all models.

Theorem 17 *The following formulae are strongly valid in the class of plausibly serial CTLKP models:*
(i) $B_a\varphi \leftrightarrow K_a\mathbf{Pl}_a\varphi$, *(ii)* $K_a\varphi \leftrightarrow B_a\mathbf{Ph}\,\varphi$.

Theorem 18 *Formula $B_a\varphi \leftrightarrow K_a\mathbf{Pl}_a(\mathsf{E}\bigcirc \top \to \varphi)$ is strongly valid.*

Note that this characterization has a strong commonsense reading: *believing in φ is knowing that φ plausibly holds in all plausibly imaginable situations*.

4.4.3 Properties of the Update

The first notable property of plausibility update is that it influences only formulae in which plausibility plays a role, i.e. ones in which belief or plausibility modalities occur.

Proposition 43 *Let φ be a CTLKP formula that does not include operators $\mathbf{Pl}_a$ and B_a, and γ be a CTLKP path formula. Then, we have $\models \varphi \leftrightarrow (\textbf{\textit{set-pl}}_a\ \gamma)\varphi$.*

What can be said about the result of an update? At first sight, formula $(\textbf{set-pl}_a\ \gamma)\mathbf{Pl}_a\mathsf{A}\gamma$ seems a natural characterization; however, it is not valid. This is because, by leaving the other (implausible) paths out of scope, we may leave out of $|\gamma|$ some paths that were needed to satisfy γ (see the example in Section 4.4.2). We propose two alternative ways out: the first one restricts the language of the update similarly to [132]; the other refers to physical possibilities, in a way analogous to [85].

Proposition 44 *The CTLKP formula $(\textbf{set-pl}_a\ \gamma)\mathbf{Pl}_a\mathsf{A}\gamma$ is not valid. However, we have the following validities:*

1. $\models (\textbf{set-pl}_a\ \gamma_u)\mathbf{Pl}_a\mathsf{A}\gamma_u$*, where γ_u is a universal CTLK path formula from Definition 7.*
2. *If $\varphi, \varphi_1, \varphi_2$ are arbitrary CTLK formulae, then:*
$$\models (\textbf{set-pl}_a\ \bigcirc\varphi)\mathbf{Pl}_a\mathsf{A}\bigcirc(\mathbf{Ph}\,\varphi),$$
$$\models (\textbf{set-pl}_a\ \Box\varphi)\mathbf{Pl}_a\mathsf{A}\Box(\mathbf{Ph}\,\varphi)\textit{, and}$$
$$\models (\textbf{set-pl}_a\ \varphi_1\,\mathcal{U}\,\varphi_2)\mathbf{Pl}_a\mathsf{A}(\mathbf{Ph}\,\varphi_1)\,\mathcal{U}\,(\mathbf{Ph}\,\varphi_2).$$

4.5 Verification of Plausibility, Time and Beliefs

In this section we report preliminary results on model checking CTLKP formulae. Clearly, verifying CTLKP properties directly against models with plausibility does not make much sense, since these models are inherently infinite; what we need is a finite representation of plausibility sets. One such representation has been discussed in Section 4.3.3: plausibility sets can be defined by path formulae and the update operator $(\textbf{set-pl}_a\ \gamma)$.

We follow this idea here, studying the complexity of model checking CTLKP formulae *against CTLK models* (which can be seen as a compact representation of CTLKP models in which all the paths are assumed plausible), with the underlying idea that plausibility sets, when needed, must be defined explicitly in the object language. Below we sketch an algorithm that model-checks CTLKP formulae in time linear wrt the size of the model and the length of the formula. This means that we have extended CTLK to a more expressive language with no computational price to pay.

First of all, we get rid of the belief operators (due to Theorem 18), replacing every occurrence of $B_a\varphi$ with $K_a\mathbf{Pl}_a(\mathsf{E}\bigcirc\top\rightarrow\varphi)$. Now, let $\overrightarrow{\gamma} = \langle\gamma_1, ..., \gamma_k\rangle$ be a vector of "vanilla" path formulae (one per agent), with the initial vector $\overrightarrow{\gamma_0} = \langle\top, ..., \top\rangle$, and $\overrightarrow{\gamma}[\gamma'/a]$ denoting vector $\overrightarrow{\gamma}$, in which $\overrightarrow{\gamma}[a]$ is replaced with γ'. Additionally, we define $\overrightarrow{\gamma}[0] = \top$. We translate the resulting CTLKP formulae to ones without plausibility via function $tr(\varphi) = tr_{\overrightarrow{\gamma_0},0}(\varphi)$, defined as follows:

$$\begin{aligned}
&tr_{\overrightarrow{\gamma},i}(p) = p,\\
&tr_{\overrightarrow{\gamma},i}(\varphi_1\wedge\varphi_2) = tr_{\overrightarrow{\gamma},i}(\varphi_1)\wedge tr_{\overrightarrow{\gamma},i}(\varphi_2),\\
&tr_{\overrightarrow{\gamma},i}(\neg\varphi) = \neg tr_{\overrightarrow{\gamma},i}(\varphi),\\
&tr_{\overrightarrow{\gamma},i}(K_a\varphi) = K_a\, tr_{\overrightarrow{\gamma},0}(\varphi),\\
&tr_{\overrightarrow{\gamma},i}(\mathbf{Pl}_a\,\varphi) = tr_{\overrightarrow{\gamma},a}(\varphi),\\
&tr_{\overrightarrow{\gamma},i}((\textbf{set-pl}_a\ \gamma')\varphi) = tr_{\overrightarrow{\gamma}[\gamma'/a],i}(\varphi),
\end{aligned}$$

$$tr_{\overrightarrow{\gamma},i}(\mathbf{Ph}\,\varphi) = tr_{\overrightarrow{\gamma},0}(\varphi),$$
$$tr_{\overrightarrow{\gamma},i}(\bigcirc\varphi) = \bigcirc tr_{\overrightarrow{\gamma},i}(\varphi),$$
$$tr_{\overrightarrow{\gamma},i}(\Box\varphi) = \Box tr_{\overrightarrow{\gamma},i}(\varphi),$$
$$tr_{\overrightarrow{\gamma},i}(\varphi_1\,\mathcal{U}\,\varphi_2) = tr_{\overrightarrow{\gamma},i}(\varphi_1)\,\mathcal{U}\,tr_{\overrightarrow{\gamma},i}(\varphi_2),$$
$$tr_{\overrightarrow{\gamma},i}(\mathsf{E}\gamma') = \mathsf{E}(\overrightarrow{\gamma}[i] \wedge tr_{\overrightarrow{\gamma},i}(\gamma')).$$

Note that the resulting sentences belong to the logic of CTLK+, that is CTL+ (where each path quantifier can be followed by a *Boolean combination* of "vanilla" path formulae)[9] with epistemic modalities. The following proposition justifies the translation.

Proposition 45 *For any CTLKP formula φ without B_a, we have that $M, q \models_{CTLKP} \varphi$ iff $M, q \models_{CTLK+} tr(\varphi)$.*

In general, model checking CTL+ (and also CTLK+) is Δ_2^P-complete. However, in our case, the Boolean combinations of path subformulae are always conjunctions of at most two non-negated elements, which allows us to propose the following model checking algorithm. First, subformulae are evaluated recursively: for every subformula ψ of φ, the set of states in M that satisfy ψ is computed and labeled with a new proposition p_ψ. Now, it is enough to define checking $M, q \models \varphi$ for φ in which all (state) subformulae are propositions, with the following cases:

Case $M, q \models \mathsf{E}(\Box p \wedge \gamma)$**:** If $M, q \not\models p$, then return **no**. Otherwise, remove from M all the states that do not satisfy p (yielding a sparser model M'), and check the CTL formula $\mathsf{E}\gamma$ in M', q with any CTL model-checker.

Case $M, q \models \mathsf{E}(\bigcirc p \wedge \gamma)$**:** Create M' by adding a copy q' of state q, in which only the transitions to states satisfying p are kept (i.e., $M, q' \models r$ iff $M, q \models r$; and $q' R q''$ iff qRq'' and $M, q'' \models p$). Then, check $\mathsf{E}\gamma$ in M', q'.

Case $M, q \models \mathsf{E}(p_1\,\mathcal{U}\,p_2 \wedge p_3\,\mathcal{U}\,p_4)$**:** Note that this is equivalent to checking $\mathsf{E}(p_1 \wedge p_3)\,\mathcal{U}\,(p_2 \wedge \mathsf{E}p_3\,\mathcal{U}\,p_4) \vee \mathsf{E}(p_1 \wedge p_3)\,\mathcal{U}\,(p_4 \wedge \mathsf{E}p_1\,\mathcal{U}\,p_2)$, which is a CTL formula.

Other cases: The above cases cover all possible formulas that begin with a path quantifier. For other cases, standard CTLK model checking can be used.

Theorem 19 *Model checking CTLKP against CTLK models is PTIME-complete, and can be done in time $O(ml)$, where m is the number of transitions in the model, and l is the length of the formula to be checked. That is, the complexity is no worse than for CTLK itself.*

4.6 Conclusions

In this paper a notion of *plausible behavior* is considered, with the underlying idea that *im*plausible options should be usually ignored in practical reasoning about possible future courses of action. We add the new notion

[9] For the semantics of CTL+, and discussion of model checking complexity, cf. [96].

of plausibility to the logic of CTLK [115], and obtain a language which enables reasoning about what can (or must) plausibly happen. As a technical device to define the semantics of the resulting logic, we use a non-standard satisfaction relation $\models_P$ that allows to propagate the "current" set of plausible paths into subformulae. Furthermore, we propose a non-standard notion of beliefs, defined in terms of indistinguishability and plausibility. We also propose how plausibility assumptions can be specified in the object language via a *plausibility update operator* (in a way similar to [85]).

We use this new framework to investigate some important properties of plausibility, knowledge, beliefs, and updates. In particular, we show that knowledge is an **S5** modality, and that beliefs satisfy axioms **K45** in general, and **KD45** for the class of *plausibly serial models*. We also prove that *believing in* φ is *knowing that* φ *plausibly holds in all plausibly possible situations*. That is, the relationship between knowledge and beliefs is very natural and reflects the initial intuition precisely. Moreover, the model checking results from Section 4.5 show that verification for CTLKP is no more complex than for CTL and CTLK.

We would like to stress that we do not see this contribution as a mere technical exercise in formal logic. Human agents use a similar concept of plausibility and "practical" beliefs in their everyday reasoning in order to reduce the search space and make the reasoning feasible. As a consequence, we suggest that the framework we propose may prove suitable for modeling, design, and analysis resource-bounded agents in general.

We would like to thank Juergen Dix for fruitful discussions, useful comments and improvements.

Chapter 5

Reasoning about Temporal Properties of Rational Play (joint work with Nils Bulling and Jürgen Dix)

Abstract. This article is about defining a suitable logic for expressing classical game theoretical notions. We define an extension of alternating-time temporal logic (ATL) that enables us to express various rationality assumptions of intelligent agents. Our proposal, the logic ATLP (*ATL with plausibility*) allows us to specify sets of rational strategy profiles in the object language, and reason about agents' play if only these strategy profiles were allowed. For example, we may assume the agents to play only Nash equilibria, Pareto-optimal profiles or undominated strategies, and ask about the resulting behaviour (and outcomes) under such an assumption. The logic also gives rise to generalized versions of classical solution concepts through characterizing patterns of payoffs by suitably parameterized formulae of ATLP. We investigate the complexity of model checking ATLP for several classes of formulae: It ranges from $\mathbf{\Delta_3^P}$ to **PSPACE** in the general case and from $\mathbf{\Delta_3^P}$ to $\mathbf{\Delta_4^P}$ for the most interesting subclasses, and roughly corresponds to solving extensive games with imperfect information.

`Keywords`: game theory, modal and temporal logic, reasoning about agents, rationality.

5.1 Introduction

Alternating-time temporal logic (ATL) [6, 8] is a temporal logic that incorporates some basic game theoretical notions. In ATL we can express that a

group of agents is able to *bring about* ψ, i.e., they are able to ensure a situation where ψ holds whatever the other agents might do. However, such a statement is weaker than it seems. Often, we know that agents behave according to some rationality assumptions, they are not completely dumb. Therefore we do not have to check *all possible plays* – only those that are *plausible* in some reasonable sense. This has striking similarities to nonmonotonic reasoning, where one considers *default rules* that describe the most plausible behaviour and allow to draw conclusions when knowledge is incomplete.

In general, plausibility can be seen as a broader notion than rationality: One may obtain plausibility specifications e.g. from learning or folk knowledge. In this article, however, we mostly focus on plausibility as rationality in a game-theoretical sense.

Our idea has been inspired by the way in which games are analyzed in game theory. Firstly, game theory identifies a number of *solution concepts* (e.g., Nash equilibrium, undominated strategies, Pareto optimality) that can be used to define rational behaviour of players. Secondly, we usually *assume that players play rationally* in the sense of one of the above concepts, and we *ask about the outcome of the game under this assumption*.

Solution concepts do not only help to determine the right decision for an agent. Perhaps more importantly, they *constrain* the possible (predicted) responses of the opponents to a proper subset of all the possibilities. For many games the number of all possible outcomes is infinite, although only some of them, often finitely many, *make sense*. We need a notion of rationality (like subgame-perfect Nash equilibrium) to discard the *less sensible* ones, and to determine what should happen had the game been played by ideal players.

5.1.1 Idea and Main Results

While ATL is already a logic that incorporates some game theoretical concepts, we claim that extending ATL by other useful constructs not only helps us to better understand the classical solution concepts in game theory, but it also paves the way for defining new solution concepts (which we call *general*). We extend ATL by the notion of *plausibility*, and call the resulting logic ATLP. We claim that this logic is suitable to model and to reason about the rational behaviour of agents.

In this article we discuss the following:

1. We recall from [14, 85] that models of ATL, called *concurrent game structures (CGS)*, embed *extensive form games with perfect information* in a natural way. This can be done, e.g., by adding auxiliary propositions to the CGS, that describe the payoffs of agents. With this perspective, concurrent game structures can be seen as a strict generalisation of extensive games.

2. We discuss informally how these more general games can be "solved", given an appropriate solution concept that defines which plays can be plausibly expected.

3. We extend ATL to a new logic ATLP that allows to reason about what agents can achieve under an arbitrary plausibility assumption. Analy-

sis of this kind typically starts with assuming that agents are rational in the sense that they only play strategies consistent with a selected solution concept (e.g., they can only play Nash equilibria, or undominated strategies etc.). Then, we can ask which outcomes can be obtained by whom under this assumption.

4. We extend the results from [130, 85], and show that the classical solution concepts (*Nash equilibrium*, *subgame perfect Nash equilibrium*, *Pareto optimality*, and others) can be also characterized in the object language of ATLP. That is, we propose expressions of ATLP that, given an extensive game, denote exactly the set of Nash equilibria (subgame perfect NE's, Pareto optimal profiles, etc.) in that game. In consequence, ATLP can serve both as a language for reasoning about rational play, and for specifying what rational play is. We point out that these characterizations extend traditional solution concepts to the more general class of multi-stage multi-player games defined by concurrent game structures.

5. We also propose an alternative approach to defining solution concepts for games that involve infinite flow of time. In the new approach, path formulae of ATL are used to specify the "winning conditions" of each player. This implicitly leads to a normal form game with binary payoffs, where the traditional solution concepts are well defined. We also demonstrate how these "qualitative" solution concepts (parametrized by ATL path formulae) can be characterized in ATLP.

6. We constructively show that several logics can be embedded into ATLP. That is, we demonstrate how models and formulae of those logics can be (independently) transformed to their ATLP counterparts in a way that preserves their truth values.

7. Last but not least, we investigate the model checking problem in ATLP. We show that, for different subclasses of the new logic, the complexity of model checking ranges from $\mathbf{\Delta_3^P}$-completeness to $\mathbf{PSPACE}$-completeness. We also argue that, when the number of plausible strategy profiles is reasonably small, the model checking can be done in polynomial time.

5.1.2 Related Work

In our approach, some strategies (or rather *strategy profiles*) can be assumed plausible, and one can reason what can be *plausibly* achieved by agents under such an assumption. There are two possible points of focus in this context. Research within game theory understandably favors work on *characterization* of various types of rationality (and defining most appropriate solution concepts). Applications of game theory, also understandably, tend toward *using* the solution concepts in order to predict the outcome in a given game (in other words, to "solve" the game).

The first issue has been studied in the framework of logic, for example in [11, 19, 124, 125]; more recently, game-theoretical solution concepts have been characterized in dynamic logic [62, 61], dynamic epistemic logic [14, 128], and ATL [130, 85].

The second thread seems to have been neglected in logic-based research: papers by Van Otterloo and his colleagues [137, 138, 136, 135] are the only exceptions we know of. Moreover, every proposal from [137, 138, 136, 135] commits to a particular view of rationality (Nash equilibria, undominated strategies etc.). In this paper, we try to generalize this kind of reasoning in a way that allows to "plug in" any solution concept of choice. We also try to fill in the gap between the two threads by showing how sets of rational strategy profiles can be specified in the object language, and building upon the existing work on modal logic characterizations of solution concepts [62, 61, 14, 128, 130, 85].

5.1.3 Structure of the Article

We begin by introducing some basic notions from game theory and the alternating-time temporal logic (Section 5.2). In Section 5.3, we pave the way for Sections 5.4 and 5.5: We relate ATL and its semantical models to extensive games. Then we do the same for an extension of ATL, called ATLI, which has been introduced in [85] to characterize solution concepts in extensive games.

Section 5.4 introduces our logic ATLP: We extend ATL with a plausibility operator. This constitutes the base language $\mathcal{L}^{\text{base}}_{ATLP}$. The main syntactic novelty are *plausibility terms* that refer to rational strategies. Then, we extend the base language by allowing to *specify sets of rational strategy profiles in the object language*. To do this, we need to define a language with a much richer structure of terms as in $\mathcal{L}^{\text{base}}_{ATLP}$. We achieve this by describing strategy profiles with ATLI formulae, and extending $\mathcal{L}^{\text{base}}_{ATLP}$ so that the concepts presented in Section 5.3.4 can be reused. Finally, we propose the full language $\mathcal{L}_{ATLP}$ where ATLP characterizations of solution concepts are "plugged" into ATLP formulae that describe the consequences of adopting this or that notion of rationality. Thus, we create a single language for both characterizing rational behaviour and reasoning about its outcome. We define $\mathcal{L}_{ATLP}$ through a hierarchy of sublanguages $\mathcal{L}^{k}_{ATLP}$, each allowing for more levels of plausibility updates than the previous one.

Section 5.5 lists our main conceptual results. We show how to embed several logics in ATLP and how to express several classical solution concepts (such as Nash equilibria and others) already in $\mathcal{L}^{1}_{ATLP}$. Our third result is the generalization of Nash equilibria, Pareto optimality, undominatedness and subgame perfect Nash equilibria as certain parameterized formulae in the language of ATLP.

Section 5.6 contains the results of our study on the complexity of model checking in variants of ATLP. Finally, we conclude with Section 5.7.

Some results reported in this article have been already presented in a preliminary form in several conference and workshop papers. A rough idea of "ATL with plausibility" was proposed in [23, 76]. In [77], we studied a more complex language of terms that would allow to specify sets of rational strategy profiles in the object language; still, the language was not expressive enough for our purposes. Some initial complexity results were also reported in that paper. Finally, [26] put forward the idea that rationality specifications can be written in ATLP itself, and nested in ATLP formulae. The idea of

$1 \backslash 2$	a_2^1	a_2^2
a_1^1	$\langle \boldsymbol{\mu}_1(a_1^1, a_2^1), \boldsymbol{\mu}_2(a_1^1, a_2^1) \rangle$	$\langle \boldsymbol{\mu}_1(a_1^1, a_2^2), \boldsymbol{\mu}_2(a_1^1, a_2^2) \rangle$
a_1^2	$\langle \boldsymbol{\mu}_1(a_1^2, a_2^2), \boldsymbol{\mu}_2(a_1^2, a_2^2) \rangle$	$\langle \boldsymbol{\mu}_1(a_1^2, a_2^1), \boldsymbol{\mu}_2(a_1^2, a_2^2) \rangle$

Figure 5.1: Payoff matrix for 2 players and 2×2 strategies

"qualitative" solution concept was also introduced in [26].

5.2 Preliminaries

In this section, we introduce some concepts that are important for the rest of this article. After recapitulating some machinery of game theory, together with two running examples, we introduce ATL, which is the basis for our new logic ATLP.

5.2.1 Concepts From Game Theory

We start with the definition of a *normal form* game, also called *strategic game*, and use the terminology of [111].

Definition 8 (Normal Form (NF) Game) *A (perfect information)* normal form game Γ*, is a tuple of the form* $\Gamma = \langle \mathcal{P}, \mathcal{A}_1, \ldots, \mathcal{A}_k, \mu \rangle$*, where*

- $\mathcal{P}$ *is a finite set of* players *(or* agents*), with* $|\mathcal{P}| = k$*,*
- $\mathcal{A}_i$ *are nonempty sets of* actions *(or* strategies*) for player* i*,*
- $\mu : \mathcal{P} \to (\prod_{i=1}^{k} \mathcal{A}_i \to \mathbb{R})$ *is the* payoff function *(which we also write* $\langle \mu_1, \ldots, \mu_k \rangle$*).*

A combinations of actions (resp. strategies, payoffs), one per player, will be called an action profile *(resp.* strategy profile*,* payoff profile*) throughout the paper.*

Such games are usually depicted with a payoff matrix. For example, a game with 2 players having 2 strategies each is represented by the matrix in Figure 5.1.

Example 16 (Classical NF Games) *Some classical NF games with* 2 *players and* 2 *strategies are shown in Figure 5.2. In the* Matching Pennies *game, player 1 wins when both pennies show the same side. Otherwise player 2 wins. In the* Prisoner's Dilemma*, two prisoners can either cooperate or defect with the police. Finally, the* Hawk-Dove *game is similar, but the payoffs are different. The higher the payoff the better it is for the respective player.*

Definition 9 (Solution Concepts in Games) *There are several well-known* solution concepts *such as:*

Nash Equilibrium (NE): *A strategy profile such that no agent can unilaterally deviate from her strategy and get a better payoff;*

1\ 2	Head	Tail
Head	(1, -1)	(-1, 1)
Tail	(-1, 1)	(1, -1)

1\ 2	C	D
C	(3, 3)	(0, 5)
D	(5, 0)	**(1, 1)**

1\ 2	Dove	Hawk
Dove	(3, 3)	**(1, 4)**
Hawk	**(4, 1)**	(0, 0)

Figure 5.2: Payoff matrices for Matching Pennies, Prisoner's Dilemma, and Hawk-Dove. Nash equilibria are set in bold font.

Pareto Optimality (PO): *There is no other strategy profile that leads to a payoff profile which is at least as good for each agent, and strictly better for at least one agent;*

Weakly Undominated Strategies (UNDOM): *These are strategies that are not dominated by any other strategy, i.e., such that there is no strategy at least as good for all the responses of the opponent, and strictly better for at least one response.*

We do not repeat the formal definitions here and refer to the literature [111]. We point out, however, that some solution concepts yield sets of individual strategies (UNDOM), while others produce rather sets of strategy profiles *(NE, PO).*

In the examples from Figure 5.2, there is no Nash equilibrium for the Matching Pennies game, exactly one Nash equilibrium for the Prisoner's Dilemma (namely, the strategy profile $\langle D, D\rangle$), and two Nash equilibria for the Hawk-Dove game ($\langle Hawk, Dove\rangle$ and $\langle Dove, Hawk\rangle$).

In NF games, agents do their moves *simultaneously*: They do not see the move of the opponent and therefore cannot act accordingly. On the other hand, there are many games where the move of one player should depend on the preceding move of the opponent, or even on the *whole history*. This idea is captured in games of *extensive form*.

Definition 10 (Extensive Form (EF) Game) *A (perfect information)* extensive form game Γ *is a tuple of the form* $\Gamma = \langle \mathcal{P}, \mathcal{A}, H, ow, u\rangle$*, where:*

- $\mathcal{P}$ *is a finite set of players,*
- $\mathcal{A}$ *a finite set of actions (moves),*
- H *is a set of finite action sequences (game histories), such that (1)* $\emptyset \in H$*, (2) if* $h \in H$*, then every initial segment of* h *is also in* H*. We use the notation* $\mathcal{A}(h) = \{m \mid h \circ m \in H\}$ *to denote the* moves available at h*, and* $Term = \{h \mid \mathcal{A}(h) = \emptyset\}$*, the set of* terminal *positions,*
- $ow : H \to \mathcal{P}$ *defines which player "owns" history* h*, i.e., has the next move given* h*,*

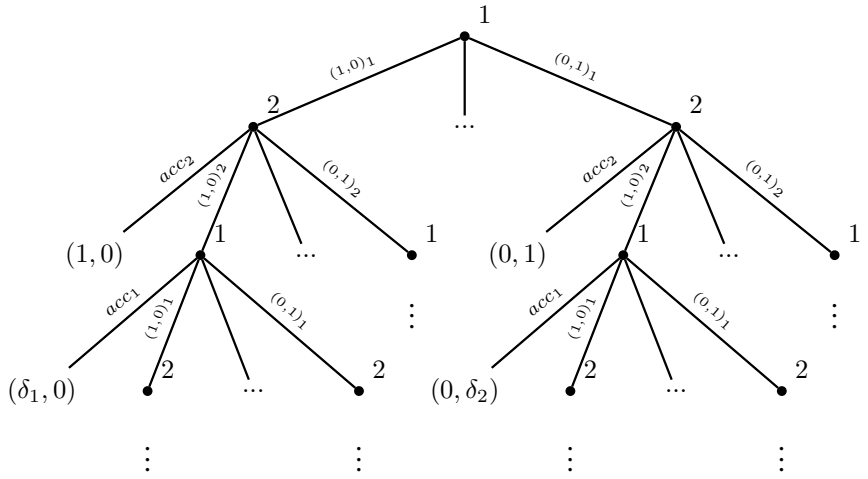


Figure 5.3: The bargaining game.

- $u : \mathcal{P} \times Term \to U$ *assigns agents' utilities to every terminal position of the game.*

We will usually assume that the set of utilities U is finite.

Such games can be easily represented as trees of all possible plays.

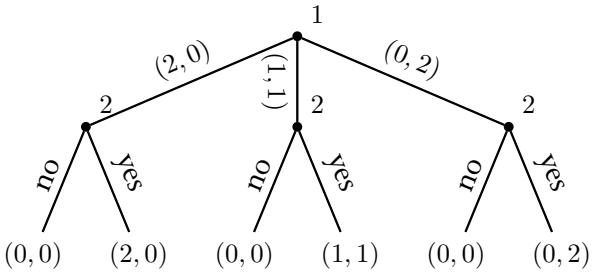
Example 17 (Bargaining) *Consider bargaining with discount [111, 120]. Two players,* 1 *and* 2*, bargain about how to split goods worth initially $w_0 = 1$ EUR. After each round without agreement, the subjective worth of the goods reduces by* discount rates δ_1 *(for player a_1) and δ_2 (for player a_2). So, after t rounds, the goods are worth $\langle \delta_1^t, \delta_2^t \rangle$, respectively. Subsequently, a_1 (if t is even) or a_2 (if t is odd) makes an offer to split the goods in proportions $\langle x, 1-x \rangle$, and the other player accepts or rejects it. If the offer is accepted, then a_1 takes $x\delta_1^t$, and a_2 gets $(1-x)\delta_2^t$; otherwise the game continues. The (infinite) extensive form game is shown in Figure 5.3. Note that the tree has infinite depth as well as an inifite branching factor.*

In order to obtain a finite set of payoffs, it is enough to assume that the goods are split with finite precision represented by a rounding function $r : \mathbb{R} \to \mathbb{R}$. So, after t rounds, the goods are in fact worth $\langle r(\delta_1^t), r(\delta_2^t) \rangle$, respectively, and if the offer is accepted, then a_1 takes $r(x\delta_1^t)$, and a_2 gets $r((1-x)\delta_2^t)$.

A *strategy* for player $i \in \mathcal{P}$ in extensive game Γ is a function that assigns a legal move to each history owned by i. Note that a *strategy profile* (i.e., a combination of strategies, one per player) determines a unique path from the game root ($\emptyset$) to one of the terminal nodes (and hence also a single profile of payoffs). In consequence, one can construct the corresponding normal from game $NF(\Gamma)$ by enumerating strategy profiles and filling the payoff matrix with resulting payoffs.

Example 18 (Sharing Game) *Consider the* Sharing Game *in Figure 5.4A. Its corresponding normal form game is presented in Figure 5.4B. Firstly, player* 1 *can suggest how to share, say, two 1 EUR coins. E.g.* $(2, 0)$ *means that* 1 *gets two*

(A)



(B)

1\ 2	nnn	nny	nyn	nyy	ynn	yny	yyn	yyy
$(2,0)$	**(0, 0)**	**(0, 0)**	(0, 0)	(0, 0)	**(2, 0)**	**(2, 0)**	**(2, 0)**	**(2, 0)**
$(1,1)$	(0, 0)	(0, 0)	**(1, 1)**	**(1, 1)**	(0, 0)	(0, 0)	(1, 1)	(1, 1)
$(0,2)$	(0, 0)	**(0, 2)**	(0, 0)	(0, 2)	(0, 0)	(0, 2)	(0, 0)	(0, 2)

Figure 5.4: The Sharing game: (A) Extensive form; (B) Normal form. Nash equilibria are set in bold font. A strategy abc ($a, b, c \in \{y, n\}$) of player 2 denotes the strategy in which 2 plays a (resp. b, c) if player 1 has played $(2,0)$ (resp. $(1,1)$, $(0,2)$) where n refers to "no" and y to "yes".

euro and 2 *gets nothing. Subsequently, player* 2 *can accept the offer or reject it; in the latter case both players get nothing.*

The game includes 3 *strategies for player* 1 *(which can be denoted by the action that they prescribe at the beginning of the game), and* 8 *strategies for player* 2 *(generated by the combination of actions prescribed for the second move), which gives* 24 *strategy profiles in total. However, not all of them seem plausible. Constraining the possible plays to Nash equilibria only, we obtain* 9 *"rational" strategy profiles (cf. Figure 5.4B), although it is still disputable if all of them really "make sense".*

A *subgame* of an extensive game Γ is defined by a subtree of the game tree of Γ.

Definition 11 (Subgame Perfect Nash Equilibrium (SPN)) *This solution concept is an extension of NE: A strategy is a SPN in* Γ *if it is a NE in* Γ *and, in addition, a NE in all subgames of* Γ.

Example 19 (Sharing Game ctd.) *Consider again the game from Example 18. While the game has* 9 *Nash equilibria, only two of them are subgame perfect:* $\langle (2,0), yyy \rangle$ *and* $\langle (1,1), nyn \rangle$.

Example 20 (Bargaining ctd.) *Consider the bargaining game from Example 22. The game has an immense number of possible outcomes. Still worse, every strategy profile*

$$s^x : \begin{cases} a_1 \textit{ always offers } \langle x, 1-x \rangle, \textit{ and agrees to } \langle y, 1-y \rangle \textit{ for } y \geq x \\ a_2 \textit{ always offers } \langle x, 1-x \rangle, \textit{ and agrees to } \langle y, 1-y \rangle \textit{ iff } 1-y \geq 1-x \end{cases}$$

is a Nash equilibrium (NE): an agreement is reached in the first round. *Thus, every split* $\langle x, 1-x\rangle$ *can be achieved through a Nash equilibrium; it seems that a stronger solution concept is needed. Indeed, the game has a unique* subgame perfect Nash equilibrium. *Because of the finite precision, there is a minimal round* T *with* $r(\delta_i^{T+1}) = 0$ *for* $i = 1$ *or* $i = 2$*. For simplicity, assume that* $i = 2$ *and agent* a_1 *is the offerer in* T *(i.e.,* T *is even). Then, the only subgame perfect NE is given by the strategy profile* s^κ *with* $\kappa = (1-\delta_2)\frac{1-(\delta_1\delta_2)^{\frac{T}{2}}}{1-\delta_1\delta_2} + (\delta_1\delta_2)^{\frac{T}{2}}$*. The goods are split* $\langle \kappa, 1-\kappa\rangle$*; the agreement is reached in the first round.*[1]

5.2.2 ATL

Alternating-time temporal logic (ATL) [6, 8] enables reasoning about temporal properties and strategic abilities of agents. Formally, the language of ATL is given as follows.

Definition 12 ($\mathcal{L}_{ATL}$) *Let* $\mathbb{A}\mathrm{gt} = \{a_1, \ldots, a_k\}$ *be a nonempty finite set of all agents, and* Π *be a set of propositions (with typical element* p*). We use the symbol* a *to denote a typical agent, and* A *to denote a typical group of agents from* $\mathbb{A}\mathrm{gt}$*. The logic* $\mathcal{L}_{ATL}(\mathbb{A}\mathrm{gt}, \Pi)$ *is defined by the following grammar:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A\rangle\!\rangle \bigcirc \varphi \mid \langle\!\langle A\rangle\!\rangle \Box\varphi \mid \langle\!\langle A\rangle\!\rangle \varphi \,\mathcal{U}\, \varphi.$$

Informally, $\langle\!\langle A\rangle\!\rangle\varphi$ says that agents A have a collective strategy to enforce φ. ATL formulae include the usual temporal operators: $\bigcirc$ (*in the next state*), $\Box$ (*always from now on*) and $\mathcal{U}$ (strict *until*). Additionally, $\Diamond$ (*now or sometime in the future*) can be defined as $\Diamond\varphi \equiv \top\,\mathcal{U}\,\varphi$. Like in CTL [29], every occurrence of a temporal operator is immediately preceded by exactly one cooperation modality (this variant of the language is sometimes called "vanilla" ATL). The broader language of ATL*, where no such restriction is imposed, is not discussed in this article. It should be noted that the CTL path quantifiers A, E can be expressed in ATL with $\langle\!\langle\emptyset\rangle\!\rangle, \langle\!\langle\mathbb{A}\mathrm{gt}\rangle\!\rangle$ respectively. The semantics of ATL is defined over *concurrent game structures*.

Definition 13 (CGS) *A* concurrent game structure (CGS) *is a tuple:* $M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o\rangle$*, consisting of: a set* $\mathbb{A}\mathrm{gt} = \{a_1, \ldots, a_k\}$ *of* agents*; a set* St *of* states*; a set* Π *of* atomic propositions*; a* valuation of propositions $\pi : St \to 2^\Pi$*; a set* Act *of* actions*. Function* $d : \mathbb{A}\mathrm{gt} \times St \to 2^{Act}$ *indicates the actions available to agent* $a \in \mathbb{A}\mathrm{gt}$ *in state* $q \in St$*. We will often write* $d_a(q)$ *instead of* $d(a, q)$*, and use* $d(q)$ *to denote the set* $d_1(q) \times \cdots \times d_k(q)$ *of* action profiles *available in state* q*. Finally,* o *is a* transition function *which maps each state* $q \in St$ *and action profile* $\overrightarrow{\alpha} = \langle \alpha_1, \ldots, \alpha_k\rangle \in d(q)$ *to another state* $q' = o(q, \overrightarrow{\alpha})$*.*

Remark 14 *In the literature on ATL, the same symbols for agents (and groups of agents) are used in the semantics and in the object language; we follow this tradition here.*

A *computation* or *path* $\lambda = q_0 q_1 \cdots \in St^\omega$ is an infinite sequence of states such that there is a transition between each q_i, q_{i+1}.We define $\lambda[i] = q_i$ to

[1] For the standard version of bargaining with discount (with the continuous set of payoffs $[0, 1]$), cf. [111, 120]. Restricting the payoffs to a finite set requires to alter the solution slightly [123, 101], see also Appendix 5.8.

denote the i-th state of λ. Λ_M denotes all paths in M. The set of all paths starting in q is given by $\Lambda_M(q)$.

Definition 14 (Strategy, outcome) *A (memoryless)* strategy *of agent a is a function $s_a : St \to Act$ such that $s_a(q) \in d_a(q)$. We denote the set of such functions by Σ_a. A* collective strategy *s_A for team $A \subseteq \mathbb{A}\mathrm{gt}$ specifies an individual strategy for each agent $a \in A$; the set of A's collective strategies is given by $\Sigma_A = \prod_{a \in A} \Sigma_a$. The set of all* strategy profiles *is given by $\Sigma = \Sigma_{\mathbb{A}\mathrm{gt}}$.*

The outcome *of strategy s_A in state q is defined as the set of all paths that may result from executing s_A from state q on: $out(q, s_A) = \{\lambda \in \Lambda_M(q) \mid \forall i \in \mathbb{N}_0 \, \exists \overrightarrow{\alpha} = \langle \alpha_1, \ldots, \alpha_k \rangle \in d(\lambda[i]) \, \forall a \in A \, (\alpha_a = s_A^a(\lambda[i]) \wedge o(\lambda[i], \overrightarrow{\alpha}) = \lambda[i+1])\}$, where s_A^a denotes agent a's part of the collective strategy s_A.*

The semantics of ATL can be given by the following clauses:

$M, q \models \mathsf{p}$ iff $\mathsf{p} \in \pi(q)$

$M, q \models \neg\varphi$ iff $M, q \not\models \varphi$

$M, q \models \varphi \wedge \psi$ iff $M, q \models \varphi$ and $M, q \models \psi$

$M, q \models \langle\!\langle A \rangle\!\rangle \bigcirc \varphi$ iff there is $s_A \in \Sigma_A$ such that $M, \lambda[1] \models \varphi$ for all $\lambda \in out(q, s_A)$

$M, q \models \langle\!\langle A \rangle\!\rangle \Box \varphi$ iff there is $s_A \in \Sigma_A$ such that $M, \lambda[i] \models \varphi$ for all $\lambda \in out(q, s_A)$ and $i \in \mathbb{N}_0$

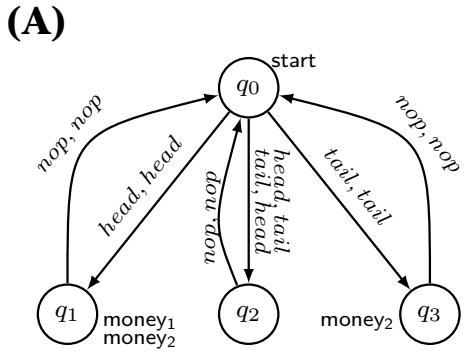
$M, q \models \langle\!\langle A \rangle\!\rangle \varphi \mathcal{U} \psi$ iff there is $s_A \in \Sigma_A$ such that, for all $\lambda \in out(q, s_A)$, there is $i \in \mathbb{N}_0$ with $M, \lambda[i] \models \psi$, and $M, \lambda[j] \models \varphi$ for all $0 \leq j < i$.

Remark 15 *We somewhat deviate from the original semantics of ATL [6, 8], where strategies assign agents' choices to* sequences *of states (which suggests that agents can recall the whole history of each game). While the choice between the two types of strategies affects the semantics of most ATL extensions, both yield equivalent semantics for* pure *ATL [121].*

5.3 Relating Games and ATL-Like Logics

In this section we present some important ideas that form the starting point for later sections. (1) We discuss informally how the notion of strategic ability in ATL can be refined so that it takes into account only "sensible" behaviour of agents. (2) We look back on the logic of GLP [138] which implements the idea formally, albeit in a very limited way. (3) We summarize a correspondence between extensive games and the models of ATL. (4) We recall an extension of ATL, called ATLI ("ATL with Intentions"), which will later serve as an intermediate logical framework and as a motivation for our logic ATLP. We also demonstrate how several game-theoretical solution concepts can be expressed in ATLI. (5) Finally we present our idea of *qualitative* solution concepts, where ATL path formulae are used to define the winning conditions.

We illustrate the ideas with two examples from the previous section: *Matching Pennies* and *Bargaining with Discounts*.



$1\backslash 2$	s_h	s_t
s_h	$1,1$	$0,0$
s_t	$0,0$	$0,1$

Figure 5.5: *Asymmetric matching pennies*: **(A)** Concurrent game structure M_1. In q_0 the agents can choose to show $head$ or $tail$. Both agents can only execute action nop (no operation) in states q_1, q_2, q_3. **(B)** The corresponding NF game. We use s_h (resp. s_t) to denote the strategy in which the player always shows head (resp. tail) in q_0 and nop in q_1, q_2, and q_3.

5.3.1 ATL and Rational Play

Example 21 (Asymmetric matching pennies) *Consider a variant of the matching pennies game, presented in Figure 5.5A. Formally, the model is given as follows:*

$$M_1 = \langle\{1,2\}, \{q_0, q_1, q_2, q_3\}, \{\mathsf{start}, \mathsf{money}_1, \mathsf{money}_2\}, \pi, \{head, tail, nop\}, d, o\rangle$$

where π is given as in the picture ($\pi(q_0) = \{\mathsf{start}\}$ etc.), $d(a, q_0) = \{head, tail\}$ for $a = 1, 2$, and $d(a, q) = \{nop\}$ for $a = 1, 2$ and $q = q_1, q_2, q_3$. The transition function o can also be read off from the picture. We use nop (no operation) as a "default" action in states q_1, q_2, and q_3 that brings the system back to the initial state. The intuition is that the game is played ad infinitum. Alternatively, one might add loops to states q_1, q_2 and q_3 to model a game that is played only once.

If both players show heads in q_0, both win a prize in the next step; if they both show tails, only player 2 wins. If they show different sides, nobody wins. Note that, e.g., $M_1, q_0 \models \langle\langle 2\rangle\rangle\Box\neg\mathsf{money}_1$, because agent 2 can play $tail$ all the time, preventing 1 from winning the prize. On the other hand, $M_1, q_0 \models \neg\langle\langle 2\rangle\rangle\Diamond\mathsf{money}_2$: Agent 2 has no strategy to guarantee that she will win.

The concurrent game structure in Figure 5.5A determines the set of available strategy profiles. However, it does not say anything about players' preferences. Suppose now that the players are only interested in getting some money sometime in the future (but it does not matter when and/or how much). The corresponding normal form game under this assumption is depicted in Figure 5.5B.

Such an analysis of the game is of course correct, yet it appears to be quite coarse. It seems natural to assume that players prefer winning money over losing it. If we additionally assume that the players are rational thinkers, it seems plausible that player 1 should always play head, as it keeps the possibility of getting money open (while playing tail guarantees loss). Under this assumption, player 2 has complete control over the outcome of the game:

She can play head too, granting herself and the other agent with the prize, or respond with tail, in which case both players lose. Note that this kind of analysis corresponds to the game-theoretical notion of *weakly dominant strategy*: For agent 1, playing head is dominant in the corresponding normal form game in Figure 5.5B, while both strategies of player 2 are undominated, so they can be in principle considered for playing.

It is still possible to refine our analysis of the game. Note that 2, knowing that 1 ought to play head and preferring to win money too, should decide to play $head$ herself. This kind of reasoning corresponds to the notion of *iterated undominated strategies*. If we assume that both players do reason this way, then $\langle s_h, s_h \rangle$ is the only rational strategy profile, and the game should end with both agents winning the prize.

5.3.2 Game Logic with Preferences

Game Logic with Preferences [138] is, to our knowledge, the only logic designed to address the outcome of rational play in games with perfect information. Here, we summarize the idea very briefly.

The central idea of GLP is facilitated by the *preference operator* $[a : \varphi]$. Interpretation of $[a : \varphi]\psi$ in model M proceeds as follows: if the truth of φ can be enforced by a, then we remove from the model all the actions of a that do *not* lead to enforcing it, and evaluate ψ in the resulting model. Thus, the evaluation of GLP formulae is underpinned by the assumption that *rational agents satisfy their preferences whenever they can*. The requirement applies to all the subtrees of the game tree, and it is called "subgame perfectness" by the authors.

The scope of GLP, however, is limited in several respects. Firstly, the models of GLP are restricted to finite game trees. Secondly, agents' preferences must be specified with propositional (non-modal) formulae, and they are evaluated only at the terminal states of the game. The temporal part of the language is limited, too. Lastly, and perhaps most importantly, the semantics of GLP is based on a very specific notion of rationality (see above). One can easily imagine variants of the semantics, in which other rationality criteria are used (NE, PO, UNDOM) to eliminate "irrational" strategies. Indeed, a preliminary version of GLP was based on the notion of Nash equilibrium rather than "subgame perfectness" [137]. In this article, we want to allow as much flexibility as possible with respect to the choice of a suitable solution concept.

5.3.3 Models of ATL vs. Extensive Games

In this section, we recall the correspondence between extensive form games and the semantical models of ATL, proposed in [85] and inspired by [14, 130].

We only consider game trees in which the set of payoffs is finite. Let U denote the set of all possible utility values in a game; U will be finite and fixed for any given game. For each value $v \in U$ and agent $a \in \mathbb{A}\mathrm{gt}$, we introduce a proposition p_a^v into our set Π, and fix $\mathsf{p}_a^v \in \pi(q)$ iff a gets payoff of at least v

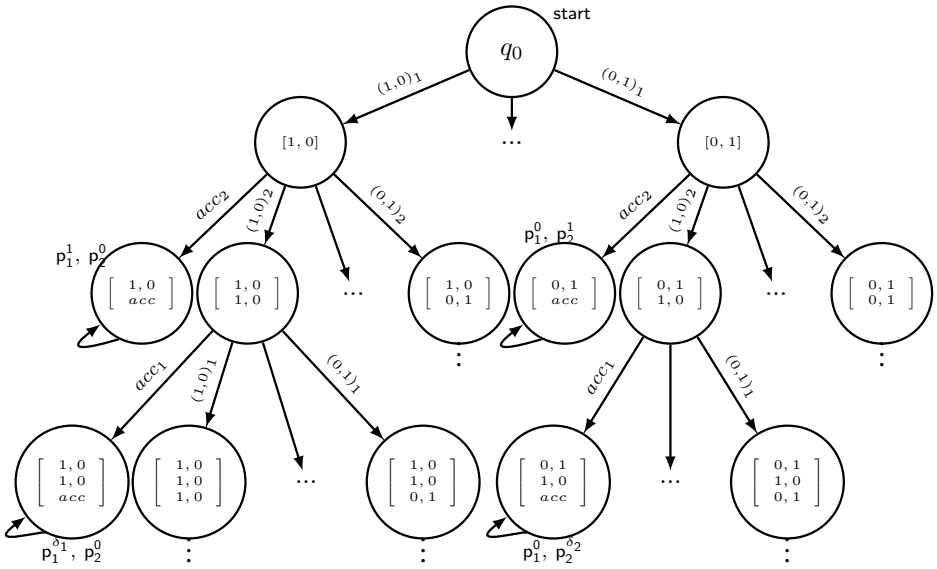


Figure 5.6: CGS M_2 for the bargaining game

in q.[2] States in the model represent finite histories in the game. In particular, we us $\emptyset$ to denote the root of the game. The correspondence between an extensive game Γ and a CGS M can be captured as follows.

Definition 15 (From Extensive Games to CGS)
A CGS $M = \{\mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o\}$ corresponds to *an extensive game* $\Gamma = \langle \mathcal{P}, \mathcal{A}, H, ow, u\rangle$ *if, and only if, the following holds:*

- $\mathbb{A}\mathrm{gt} = \mathcal{P}$*,*
- $St = H$*,*
- Π *and* π *include propositions* p_a^v *to emulate utilities for terminal states in the way described above,*
- $Act = \mathcal{A} \cup \{nop\}$*,*
- $d_a(q) = \mathcal{A}(q)$ *if* $a = ow(q)$ *and* $d_a(q) = \{nop\}$ *otherwise,*
- $o(q, nop, \ldots, m, \ldots, nop) = q \cdot m$*, and*
- $o(q, nop, nop, \ldots, nop) = q$ *for* $q \in Term$*.*

We use $M(\Gamma)$ *to refer to the CGS which corresponds to* Γ*.*

Example 22 (Bargaining in a CGS) *We consider the bargaining game from Example 17, but this time as a model of ATL. The CGS corresponding to the game shown in Figure 5.6. Nodes represent various states of the negotiation process, and arcs show how agents' moves change the state of the game. A node label refers to*

[2] Note that a state labeled by p_a^v is also labeled by $\mathsf{p}_\mathsf{a}^{v'}$ for all $v' \in U$ where $v' < v$.

the history of the game for better readability. For instance, $\left[\begin{smallmatrix} 0,1 \\ 1,0 \\ acc \end{smallmatrix}\right]$ *has the meaning that in the first round* 1 *offered* $\langle 0,1\rangle$ *which was rejected by* 2*. In the next round* 2*'s offer* $\langle 1,0\rangle$ *has been accepted by* 1 *and the game has ended.*

Note that, for every extensive game Γ, there is a corresponding CGS, but the reverse is not true: Concurrent game structures can include cycles and simultaneous moves of players, which are absent in game trees. Note also that, for those CGS's that correspond to some EF game, we get an implicit correspondence to a normal form game. We will extend this notion of correspondence to all CGS's in Section 5.3.5.

5.3.4 ATLI and Solution Concepts

The correspondence between extensive games and (some) concurrent game structures gives us a way of performing game-theoretical analysis on the latter. In particular, game-theoretical solution concepts become meaningful for these CGS's. This section illustrates how several important notions of rationality from game theory, e.g. Nash equilibria (NE), subgame perfect NE, Pareto optimality etc. can be characterized in a suitable logical language. We use the analysis from [85] where an extension of ATL, called ATLI, was employed for this purpose. We will later show how these characterizations can be "plugged" into our new logic ATLP so that one can reason about the outcome of rational play in a precisely defined sense.

We also point out after [85] that these characterizations give rise to generalized versions of solution concepts which can be applied to *all* CGS's, and not only to those that correspond to some extensive form game.

Alternating-time temporal logic with intentions (ATLI) extends ATL with formulae $(\mathbf{str}_a\, \sigma_a)\varphi$ with the intuitive reading: *Suppose that player* a *intends to play according to strategy* σ_a*, then* φ *holds*. Thus, it allows to refer to agents' strategies explicitly via terms σ_a. Let $\mathfrak{Str} = \bigcup_{a\in\mathbb{A}\mathrm{gt}} \mathfrak{Str}_a$ be a finite set of *strategic terms*. $\mathfrak{Str}_a$ are used to denote *individual strategies* of agent $a \in \mathbb{A}\mathrm{gt}$; we assume that all $\mathfrak{Str}_a$ are disjoint.

Definition 16 ($\mathcal{L}_{ATLI}$) *Let* $p \in \Pi$*,* $a \in \mathbb{A}\mathrm{gt}$*,* $A \subseteq \mathbb{A}\mathrm{gt}$*, and* $\sigma_a \in \mathfrak{Str}_a$*. The language* $\mathcal{L}_{ATLI}(\mathbb{A}\mathrm{gt}, \Pi, \mathfrak{Str})$ *is defined as:*

$$\theta ::= \mathsf{p} \mid \neg\theta \mid \theta \wedge \theta \mid \langle\!\langle A\rangle\!\rangle \bigcirc \theta \mid \langle\!\langle A\rangle\!\rangle \Box\theta \mid \langle\!\langle A\rangle\!\rangle \theta\, \mathcal{U}\, \theta \mid (\mathbf{str}_a\, \sigma_a)\theta.$$

ATLI Models $M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o, \mathcal{I}, \mathfrak{Str}, [\cdot]\rangle$ extend concurrent game structures with *intention relations* $\mathcal{I} \subseteq St \times \mathbb{A}\mathrm{gt} \times Act$ (where $q\mathcal{I}_a\alpha$ means that a possibly intends to do action α when in q). Moreover, strategic terms are interpreted as strategies according to function $[\cdot] : \mathfrak{Str} \to \bigcup_{a\in\mathbb{A}\mathrm{gt}} \Sigma_a$ such that $[\sigma_a] \in \Sigma_a$ for $\sigma_a \in \mathfrak{Str}_a$ (remember that Σ_a denotes the set of a's strategies). The set of paths consistent with all agents' intentions is defined as

$$\Lambda^{\mathcal{I}} = \{\lambda \in \Lambda_M \mid \forall i\, \exists \alpha \in d(\lambda[i])\ (o(\lambda[i], \alpha) = \lambda[i+1] \wedge \forall a \in \mathbb{A}\mathrm{gt}\ \lambda[i]\mathcal{I}_a\alpha_a)\}$$

We impose on $\mathcal{I}$ the natural requirement that $q\mathcal{I}_a\alpha$ implies that $\alpha \in d_a(q)$ for $a \in \mathbb{A}\mathrm{gt}$; that is, agents only intend to do actions if they are actually able to perform them.

We say that strategy s_A is *consistent with* A*'s intentions* if $q\mathcal{I}_a s^a_A(q)$ for all $q \in St, a \in A$. The *intention-consistent outcome set* is defined as: $out^{\mathcal{I}}(q, s_A) =$

$out(q, s_A) \cap \Lambda^{\mathcal{I}}$. The semantics of strategic operators in ATLI extends and replaces the semantic rules of ATL as follows:

$M, q \models \langle\langle A\rangle\rangle \bigcirc \theta$ iff there is a collective strategy s_A consistent with A's intentions, such that for every $\lambda \in out^{\mathcal{I}}(q, s_A)$, we have that $M, \lambda[1] \models \theta$;

$M, q \models \langle\langle A\rangle\rangle \Box\theta$ and $M, q \models \langle\langle A\rangle\rangle \theta\, \mathcal{U}\, \theta'$: analogous;

$M, q \models (\mathbf{str}_a\, \sigma)\theta$ iff $revise(M, a, [\sigma]), q \models \theta$.

The function $revise(M, a, s_a)$ updates model M by setting a's intention relation to

$$\mathcal{I}'_a = \{\langle q, s_a(q)\rangle \mid q \in St\},$$

so that s_a and $\mathcal{I}_a$ represent the same mapping in the resulting model. Note that a pure CGS M can be seen as a CGS with the full intention relation

$$\mathcal{I}^0 = \{\langle q, a, \alpha\rangle \mid q \in St, a \in \mathbb{A}\mathrm{gt}, \alpha \in d_a(q)\}.$$

Additionally, for $A = \{a_{i_1}, \ldots, a_{i_r}\}$ and $\sigma_A = \langle \sigma_{a_{i_1}}, \ldots, \sigma_{a_{i_r}}\rangle$, we define: $(\mathbf{str}_A\, \sigma_A)\varphi \equiv (\mathbf{str}_{a_{i_1}}\, \sigma_{a_{i_1}}) \ldots (\mathbf{str}_{a_{i_r}}\, \sigma_{a_{i_r}})\varphi$. Furthermore, for $B = \{b_1, \ldots, b_l\} \subseteq A$ we use $\sigma_A[B]$ to refer to B's substrategy, i.e. to $\langle \sigma_{b_1}, \ldots, \sigma_{b_l}\rangle$

Example 23 (Asymmetric matching pennies ctd.) *Coming back to our matching pennies example from Figure 5.5, we have for instance that* $M_1, q_0 \models (\mathbf{str}_1\, \sigma)\langle\langle 2\rangle\rangle \Diamond \mathsf{money}_2$ *if the denotation of* σ *is set to* s_h.

With temporal logic, it is natural to define outcomes of strategies via properties of resulting paths rather than single states. The notion of *temporal T-Nash equilibrium*, parameterized with a unary operator $T = \bigcirc, \Box, \Diamond, \_\,\mathcal{U}\,\psi, \psi\,\mathcal{U}\,\_$, was proposed in [85]. Let $\sigma = \langle \sigma_1, \ldots, \sigma_k\rangle$ be a profile of strategic terms, and let T stand for any of the following operators: $\bigcirc, \Box, \Diamond, \_\,\mathcal{U}\,\psi, \psi\,\mathcal{U}\,\_$ and let a be an agent. Then we consider the following $\mathcal{L}_{ATLI}$ formulae:

$$\begin{aligned} BR_a^T(\sigma) &\equiv (\mathbf{str}_{\mathbb{A}\mathrm{gt}\setminus\{a\}}\, \sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]) \bigwedge_{v \in U} \Big((\langle\langle a\rangle\rangle T \mathsf{p}_\mathsf{a}^\mathsf{v}) \rightarrow ((\mathbf{str}_a\, \sigma[a])\langle\langle\emptyset\rangle\rangle T \mathsf{p}_\mathsf{a}^\mathsf{v}) \Big) \\ NE^T(\sigma) &\equiv \bigwedge_{a \in \mathbb{A}\mathrm{gt}} BR_a^T(\sigma) \\ SPN^T(\sigma) &\equiv \langle\langle\emptyset\rangle\rangle \Box NE^T(\sigma). \end{aligned}$$

$BR_a^T(\sigma)$ refers to $\sigma[a]$ being a T-best strategy for a against $\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]$; $NE^T(\sigma)$ expresses that strategy profile σ is a T-Nash equilibrium; finally, $SPN^T(\sigma)$ defines σ as subgame perfect T-NE. Thus, we have a family of equilibria: $\bigcirc$-Nash equilibrium, $\Box$-Nash equilibrium etc., each corresponding to a *different temporal pattern* of utilities. For example, we may assume that *agent* a *gets* v if a utility of at least v is guaranteed for every time moment ($\Box \mathsf{p}_\mathsf{a}^\mathsf{v}$), is eventually achieved ($\Diamond \mathsf{p}_\mathsf{a}^\mathsf{v}$), and so on.

The correspondence between solution concepts and their temporal counterparts for extensive games is captured by the following proposition.

Proposition 46 *Let* Γ *be an extensive game. Then the following holds:*

1. $M(\Gamma), \emptyset \models NE^{\Diamond}(\sigma)$ *iff* $[\sigma]_{M(\Gamma)}$ *is a NE in* Γ *[85].*[3]

2. $M(\Gamma), \emptyset \models SPN^{\Diamond}(\sigma)$ *iff* $[\sigma]_{M(\Gamma)}$ *is a SPN in* Γ.

Proof sketch

1. Since $M(\Gamma)$ corresponds to an EF game, the "payoff" propositions $\mathsf{p}_{\mathsf{a}}^{\mathsf{v}}$ can only become true at the end of each path in $M(\Gamma)$. Thus, $BR_a^{\Diamond}(\sigma)$ in $M(\Gamma), \emptyset$ holds iff, whenever a can achieve the payoff of *at least* v against $\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]$ (by any strategy), it can also achieve that by using $\sigma[a]$. That is, a cannot obtain a better payoff by unilaterally changing her strategy.

2. $M(\Gamma), \emptyset \models SPN^{\Diamond}(\sigma)$ iff $M(\Gamma), q \models NE^{\Diamond}(\sigma)$ for every q reachable from the root $\emptyset$ (*). However, Γ is a tree, so every node is reachable from $\emptyset$ in $M(\Gamma)$. So, by the first part, (*) iff σ denotes a Nash equilibrium in every subtree of Γ.

■

We can use the above ATLI formulae to express game-theoretical properties of strategies in a straightforward way.

Example 24 (Bargaining ctd.) *We extend the CGS in Figure 5.6 to a CGS with intentions; then, we have* $M_2, q_0 \models NE^{\Diamond}(\sigma)$*, with* σ *interpreted in* M_2 *as* s^x *(for any* $x \in [0,1]$*). Still,* $M_2, q_0 \models SPN^{\Diamond}(\sigma)$ *if, and only if,* $[\sigma]_{M_2} = s^{\kappa}$.

We also propose a tentative ATLI characterization of *Pareto optimality* (based on the characterization from [130] for normal form games):

$$PO^{T}(\sigma) \equiv \bigwedge_{v_1} \cdots \bigwedge_{v_k} \Big((\langle\!\langle \mathbb{A}\mathrm{gt} \rangle\!\rangle T \bigwedge_{i} \mathsf{p}_{\mathsf{i}}^{\mathsf{v}_i}) \rightarrow (\mathbf{str}_{\mathbb{A}\mathrm{gt}}\, \sigma)\big((\langle\!\langle \emptyset \rangle\!\rangle T \bigwedge_{i} \mathsf{p}_{\mathsf{i}}^{\mathsf{v}_i}) \vee (\bigvee_{i} \bigvee_{\substack{v' \text{ s.t.} \\ v' > v_i}} \langle\!\langle \emptyset \rangle\!\rangle T \mathsf{p}_{\mathsf{i}}^{\mathsf{v}'}) \big) \Big).$$

That is, the strategy profile denoted by σ is Pareto optimal iff, for every achievable pattern of payoff profiles, either it can be achieved by σ, or σ obtains a strictly better payoff pattern for at least one player. Note that the above formula has exponential length with respect to the number of payoffs in U. Moreover, it is not obvious that this characterization is the right one, as it refers in fact to the evolution of payoff *profiles* (i.e., combinations of payoffs achieved by agents at the same time), and not temporal patterns of payoff evolution for *each* agent separately. So, for example, $PO^{\Diamond}(\sigma)$ may hold even if there is a strategy profile σ' that makes each agent achieve eventually a better payoff, as long as not all of them will achieve these better payoffs at the same moment. Still, the following holds.

Proposition 47 *Let* Γ *be an extensive game. Then:*

$$M(\Gamma), \emptyset \models PO^{\Diamond}(\sigma) \text{ iff } [\sigma]_{M(\Gamma)} \text{ is Pareto optimal in } \Gamma.$$

[3] The empty history $\emptyset$ denotes the root of the game tree.

Proof Let $M(\Gamma), \emptyset \models PO^{\diamond}(\sigma)$. Then, for every payoff profile $\langle v_1, \ldots, v_k \rangle$ reachable in Γ, we have that either $[\sigma]$ obtains at least as good a profile,[4] or it obtains an incomparable payoff profile. Thus, $[\sigma]$ is Pareto optimal. The proof for the other direction is analogous. ■

Example 25 (Asymmetric matching pennies ctd.) *Let M_1' be our matching pennies model M_1 with additional propositions $\mathsf{p}_\mathsf{i}^1 \equiv \mathsf{money}_\mathsf{i}$ (so, we assign to $\mathsf{money}_\mathsf{i}$ a utility of 1 for i). Then, we have $M_1', q_0 \models PO^{\diamond}(\sigma)$ iff σ denotes the strategy profile $\langle s_h, s_h \rangle$.*

5.3.5 General Solution Concepts

In this part we present an abstract formulation of our notion of *general solution concept*. We will elaborate on it later in Section 5.5.3, using our logic ATLP.

We have seen in Section 5.3.3 that some (but not all!) concurrent game structures can be seen as extensive form games, which in turn defines their correspondence to NF games. These CGS's must be turn-based (i.e., players play by taking turns) and have a tree-like structure; moreover, they must include special propositions that emulate payoffs and can be used to define agents' preferences. Now, we want to extend the correspondence to arbitrary CGS's. Our idea is to *determine the outcome of a game by the truth of certain path formulae* (e.g., in the case of binary payoffs, we can see the formulae as *winning conditions*). So, we give up the idea of assigning payoffs to leaves in a tree. Instead, we see a concurrent game structure as a game, paths in the structure as plays in the game, and satisfaction of some pre-specified formulae as the mechanism that defines agents' outcome for a given play.

Which formulae can be used in this respect?

Definition 17 (ATL Path Formulae) *By* ATL path formulae*, we denote arbitrary ATL formulae that are preceded by a temporal operator $\bigcirc, \Box, \mathcal{U}$.*

Given a CGS M and a path λ in M, satisfaction of path formulae is defined as follows:

$M, \lambda \models \bigcirc \varphi$ *iff* $M, \lambda[1] \models \varphi$

$M, \lambda \models \Box \varphi$ *iff* $M, \lambda[i] \models \varphi$ *for all* $i \in \mathbb{N}_0$

$M, \lambda \models \varphi \mathcal{U} \psi$ *iff there is* $i \in \mathbb{N}_0$ *with* $M, \lambda[i] \models \psi$*, and* $M, \lambda[j] \models \varphi$ *for all* $0 \leq j < i$.

We propose that player i's preferences can be specified by a finite list of path formulae $\eta_i = \langle \eta_i^1, \ldots, \eta_i^{n_i} \rangle$ (where $n_i \in \mathbb{N}$) with the underlying assumption that agent i prefers η_i^1 most, η_i^2 comes second best etc., and the worst outcome occurs when no $\eta_i^1, \ldots, \eta_i^{n_i}$ holds for the actual play. Thus, η_i imposes a total order on paths in a CGS.

For k players, we need a k-vector of such preference lists $\overrightarrow{\eta} = \langle \eta_1, \ldots, \eta_k \rangle$. Then, every concurrent game structure gives rise to the strategic game defined as below.

[4] We recall that $\bigwedge_i \mathsf{p}_\mathsf{i}^{\mathsf{v_i}}$ means that each player i gets *at least* v_i.

Definition 18 (From CGS To NF Game) *Let M be a CGS, $q \in St_M$ a state, and $\overrightarrow{\eta} = \langle \eta_1, \dots, \eta_k \rangle$ a vector of lists of ATL path formulae, where $k = |\mathbb{A}\mathrm{gt}|$.*

Then we define $\mathcal{S}(M, \overrightarrow{\eta}, q)$, the NF game associated with *M, $\overrightarrow{\eta}$,* and *q, as the strategic game $\langle \mathbb{A}\mathrm{gt}, \mathcal{A}_1, \dots, \mathcal{A}_k, \mu \rangle$, where the set $\mathcal{A}_i$ of i's strategies is given by Σ_i for each $i \in \mathbb{A}\mathrm{gt}$, and the payoff function is defined as follows:*

$$\mu_i(a_1, \dots, a_k) = \begin{cases} n_i - j + 1 & \text{if } \eta_i^j \text{ is the first formula from } \eta_i \text{ such that} \\ & M, \lambda \models \eta_i^j \text{ for all } \lambda \in out(q, \langle a_1, \dots, a_k \rangle), \\ 0 & \text{no } \eta_i^j \text{ is satisfied} \end{cases}$$

where $\eta_i = \langle \eta_i^1, \dots, \eta_i^{n_i} \rangle$, $1 \le j \le n_i$ and we write μ_i for $\mu(i)$.

Below, we present the generalized version of temporal Nash equilibrium and temporal subgame perfect NE.

$$\begin{aligned} BR_a^{\overrightarrow{\eta}}(\sigma) &\equiv (\mathbf{str}_{\mathbb{A}\mathrm{gt}\setminus\{a\}}\, \sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]) \bigwedge_j \Big((\langle\!\langle a \rangle\!\rangle \eta_a^j) \to ((\mathbf{str}_a\, \sigma[a]) \bigvee_{r \le j} \langle\!\langle \emptyset \rangle\!\rangle \eta_a^r) \Big) \\ NE^{\overrightarrow{\eta}}(\sigma) &\equiv \bigwedge_{a \in \mathbb{A}\mathrm{gt}} BR_a^{\overrightarrow{\eta}}(\sigma) \\ SPN^{\overrightarrow{\eta}}(\sigma) &\equiv \langle\!\langle \emptyset \rangle\!\rangle \Box NE^{\overrightarrow{\eta}}(\sigma). \end{aligned}$$

The case with a single "winning condition" per agent is particularly interesting. Clearly, it gives rise to a normal form game with binary payoffs (cf., for instance, our informal discussion of the "matching pennies" variant in Example 21). We will stick to such binary games throughout the rest of the paper (especially in Section 5.5.3 where general solution concepts are studied in more detail), but one can easily imagine how the binary case extends to the case with multiple levels of preference.

5.4 The Logic ATLP

Agents have limited ability to predict the future. However, some lines of action seem often more sensible or realistic than others. If a rationality criterion is available, we obtain means to focus on a proper subset of possible plays. In game theoretic terms, *we solve the game*, i.e., we determine the most plausible plays, and compute their outcome. In game theory, the outcome consists of the payoffs (or utilities) assigned to players at the end of the game. In temporal logics, the outcome of a play can be seen in terms of temporal patterns that can occur — which allows for much subtler descriptions. In Section 5.3.4 we explained how rationality can be characterized with formulae of modal logic (ATLI in this case). Now we show how the outcome of rational play can be described with a similar (but richer) logic, and that both aspects can be seamlessly combined.

Our logic ATLP ("ATL with Plausibility") comes in several steps, based on different underlying languages:

$\mathcal{L}_{ATLP}^{\mathbf{base}}$: Sets of plausible/rational strategy profiles can be only referred to via atomic plausibility terms (constants) whose interpretation is "hard-wired" in the model. A typical $\mathcal{L}_{ATLP}^{\mathrm{base}}$ statement is $(\mathbf{set\text{-}pl}\ \ \omega)\mathbf{Pl}\,\varphi$:

Suppose that the set of rational strategy profiles is defined by ω – then, it is plausible to expect that φ holds. For instance, one can reason about what should happen if only Nash equilibria were played, or about the abilities of players who play only Pareto optimal profiles, had terms for NE and PO been included in the model.

$\mathcal{L}^0_{ATLP}$**:** A mild extension of $\mathcal{L}^{\text{base}}_{ATLP}$. We allow some combinations of the constants of $\mathcal{L}^{\text{base}}_{ATLP}$ to form more complex terms.

$\mathcal{L}_{ATLP^{ATLI}}$**:** An intermediate language, where rational strategy profiles are characterized by ATLI formulae.

$\mathcal{L}^k_{ATLP}$**:** Here we have nestings of plausibility updates up to level k. It turns out that $\mathcal{L}_{ATLP^{ATLI}}$ is already embedded in $\mathcal{L}^1_{ATLP}$.

$\mathcal{L}_{ATLP}$**:** Unbounded nestings of formulae are allowed.

The language $\mathcal{L}^{\text{base}}_{ATLP}$ is presented in Sections 5.4.1 and 5.4.2. Then, in Section 5.4.3, we consider an intermediate step, namely plausibility terms written in ATLI. They serve as a motivation to extend $\mathcal{L}^{\text{base}}_{ATLP}$ to $\mathcal{L}^1_{ATLP}$, and, more generally, to a hierarchy $\mathcal{L}_{ATLP} = \lim_{k\to\infty} \mathcal{L}^k_{ATLP}$ which we investigate in Section 5.4.4.

5.4.1 The Language $\mathcal{L}^{\text{base}}_{ATLP}$

We extend the language of ATL with operators $\mathbf{Pl}_A$, (**set-pl** ω), and (**refn-pl** ω). The first assumes plausible behaviour of agents in A; the latter are used to fix the actual meaning of plausibility by *plausibility terms* ω. As yet, the terms are simply constants with no internal structure. Their meaning will be given later by a denotation function linking plausibility terms to sets of strategy profiles.

Definition 19 ($\mathcal{L}^{\text{base}}_{ATLP}$) *The base language $\mathcal{L}^{base}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \Omega)$ is defined over nonempty sets: Π of propositions , $\mathbb{A}\text{gt}$ of agents, and Ω of plausibility terms. We use p, a, ω to refer to typical elements of $\Pi, \mathbb{A}\text{gt}, \Omega$ respectively, and A to refer to a group of agents. $\mathcal{L}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \Omega)$ consists of all formulae defined by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle \bigcirc \varphi \mid \langle\langle A\rangle\rangle \Box \varphi \mid \langle\langle A\rangle\rangle \varphi \,\mathcal{U}\, \varphi \mid \mathbf{Pl}_A\, \varphi \mid (\textbf{\textit{set-pl}}\ \omega)\varphi \mid (\textbf{\textit{refn-pl}}\ \omega)\varphi,$$

Additionally, we define $\Diamond\varphi$ as $\top\,\mathcal{U}\,\varphi$, **Pl** *as* $\mathbf{Pl}_{\mathbb{A}\text{gt}}$*, and* **Ph** *as* $\mathbf{Pl}_{\emptyset}$*. We will often use $\mathcal{L}^{base}_{ATLP}$ to refer to the language if the sets are clear from the context.*

$\mathbf{Pl}_A$ assumes that agents in A play rationally; this means that the agents can only use strategy profiles that are *plausible* in the given model. In particular, **Pl** ($\equiv \mathbf{Pl}_{\mathbb{A}\text{gt}}$) imposes rational behaviour on all agents in the system. Similarly, **Ph** disregards plausibility assumptions, and refers to all *physically* available scenarios. The model update operator (**set-pl** ω) allows to define (or redefine) the set of plausible strategy profiles (referred to by Υ in the model) to the ones described by plausibility term ω (in this sense, it implements *revision* of plausibility). Operator (**refn-pl** σ) enables *refining* the set

of plausible strategy profiles, i.e. selecting a subset of the previously plausible profiles.

With ATLP, we can for example say that $\mathbf{Pl}\,\langle\langle\emptyset\rangle\rangle\Box(\mathsf{closed} \wedge \mathbf{Ph}\,\langle\langle guard\rangle\rangle\bigcirc \neg\mathsf{closed})$: *It is plausible to expect that the emergency door will always remain closed, but the guard retains the physical ability to open it*; or $(\mathbf{set\text{-}pl}\ \omega_{NE})\mathbf{Pl}\,\langle\langle 2\rangle\rangle\Diamond\mathsf{money}_2$: *Suppose that only playing Nash equilibria is rational; then, agent a can plausibly reach a state where she gets some money*.

We note that, in contrast to [46, 126, 24], the concept of plausibility presented in this article is *objective*, i.e. it does not vary from agent to agent. This is very much in the spirit of game theory, where rationality criteria are used in an analogous way. Moreover, it is *global*, because plausibility sets do not depend on the state of the system. Note, however, that the denotation of plausibility terms depends on the actual state.

5.4.2 Semantics of $\mathcal{L}^{\mathbf{base}}_{ATLP}$

To define the semantics of ATLP, we extend CGS's to *concurrent game structures with plausibility*. Apart from an actual set of plausible strategies Υ, a *concurrent game structure with plausibility* (CGSP) must specify the denotation of plausibility terms $\omega \in \Omega$. It is defined via a *plausibility mapping*

$$[\![\cdot]\!] : St \to (\Omega \to 2^{\Sigma})$$

Instead of $[\![q]\!](\omega)$ we will often write $[\![\omega]\!]^q$ to turn the focus to the plausibility terms. Each term is mapped to a *set* of strategy profiles. Note also, that the denotation of a term depends on the state. In a way, the current state of the system defines the "initial position in the game", and this heavily influences the set of rational strategy profiles for most rationality criteria. For example, a strategy profile can be a Nash equilibrium (NE) in q_0, and yet it may not be a NE in some of its successors.

We will propose a more concrete (and more practical) implementation of plausibility terms in Section 5.4.4.

Definition 20 (CGSP) *A* concurrent game structure with plausibility (CGSP) *is given by a tuple*

$$M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o, \Upsilon, \Omega, [\![\cdot]\!]\rangle$$

where $\langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o\rangle$ is a CGS, $\Upsilon \subseteq \Sigma$ is a set of plausible strategy profiles (called plausibility set*); Ω is a set of of plausibility terms, and $[\![\cdot]\!]$ is a plausibility mapping over St and Ω.*

By $CGSP(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ we denote the set of all CGSP's over $\mathbb{A}\mathrm{gt}$, Π and Ω. Furthermore, for a given CGSP M we use X_M to refer to element X of M, e.g., St_M to refer to the set St of states of M.

Definition 21 (Compatible model) *Given a formula $\varphi \in \mathcal{L}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ a CGSP M is called* compatible *with φ if, and only if, $M \in CGSP(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$. That is, the model interprets all symbols occurring in φ. A model M is called compatible with a set $\mathcal{L}$ of ATLP formulae if, and only if, M is compatible with each formula in $\mathcal{L}$.*

We will assume by default that, given a formula or a set of formulae, the model we consider is compatible with it.

The formula $\mathbf{Pl}\langle\langle A\rangle\rangle\gamma$ implies that A can only play plausible strategies. Thus, A's part of the strategy profiles in Υ is of particular interest which motivates the following definition.

Definition 22 (Substrategy) *Let $A, B \subseteq \mathbb{A}\mathrm{gt}$ be groups of agents such that $A \subseteq B$ and let $s_B \in \Sigma_B$ be a collective strategy for agents B. We use $s_B|_A$ to denote A's* substrategy *t_A contained in s_B , i.e., strategy $t_A \in \Sigma_A$ such that $t_A^a = s_B^a$ for every $a \in A$.[5] For a singleton coalition $\{a\}$, we also write $s_B|_a$ instead of $s_B|_{\{a\}}$.*

For a given set $P_B \subseteq \Sigma_B$ of collective strategies of agents B, $P_B|_A$ denotes the set of A's substrategies in P_B, i.e.:

$$P_B|_A = \{s_A \in \Sigma_A \mid \exists s'_B \in P_B \quad (s'_B|_A = s_A)\}.$$

Often, we impose restrictions only on a subset $B \subseteq \mathbb{A}\mathrm{gt}$ of agents, without assuming rational play of all agents. This can be desirable due to several reasons. It might, for example, be the case that only information about the proponents' play is available; hence, assuming plausible behavior of the opponents is neither sensible nor justified. Or, even simpler, a group of (simple minded) agents might be known to not behave rationally.

Consider formula $\mathbf{Pl}_B\langle\langle A\rangle\rangle\gamma$: The team A looks for a strategy that brings about γ, but the members of the team who are also in B can only choose plausible strategies. The same applies to A's opponents that are contained in B. Strategies which comply with B's part of some plausible strategy profile are called B-plausible.

Definition 23 (B-plausibility of strategies) *Let $A, B \subseteq \mathbb{A}\mathrm{gt}$ and $s_A \in \Sigma_A$. We say that s_A is* B-plausible *in M if, and only if, B's substrategy in s_A is part of some plausible strategy profile in M, i.e., if $s_A|_{A\cap B} \in \Upsilon_M|_{A\cap B}$.*

By $\Upsilon_M(B)$ we denote the set of all B-plausible strategy profiles in M. That is, $\Upsilon_M(B) = \{s \in \Sigma \mid s|_B \in \Upsilon_M|_B\}$. Note that s_A is B-plausible iff $s_A \in \Upsilon_M(B)|_A$.

We observe that s_A is trivially B-plausible whenever A and B are disjoint.

As mentioned above, if some opponents belong to the set of agents who are assumed to play plausibly then they must also comply with the actual plausibility specifications when choosing their actions; this is taken into account by the following notion of plausible outcome.

Definition 24 (B-plausible outcome) *The* B-plausible outcome, *$out_M(q, s_A, B)$, with respect to strategy s_A and state q is defined as the set of paths which can occur when only B-plausible strategy profiles can be played and agents in A follow s_A:*

$$out_M(q, s_A, B) \quad = \quad \{\lambda \in \Lambda_M(q) \mid \textit{there exists a } B\textit{-plausible } t \in \Sigma \textit{ such that } t|_A = s_A \textit{ and } out_M(q,t) = \{\lambda\}\}.$$

Note that the outcome $out_M(q, s_A, B)$ is empty whenever the $(A \cap B)$'s part of s_A is not part of any plausible strategy profile in Υ_M. For example, assume that all agents in B play only parts of Nash equilibria. Then for a given s_A there are two possibilities for the B-consistent outcome. Either it is empty because $(A \cap B)$'s part of s_A does not belong to any Nash equilibrium,

[5] We recall that s_B^a (resp. t_A^a) denotes a's part of s_B (resp. t_A).

or it consists of all paths which can occur when (1) A stick to s_A, (2) B (including $A \cap B$) play according to some Nash equilibrium, and (3) the other agents behave arbitrarily.

The truth of ATLP formulae is given with respect to a model, a state, and a set B of agents. The intuitive reading of $M, q \models_B \varphi$ is: "φ is true in model M and state q if it is assumed that players in B play rationally", i.e., by using only plausible combinations of strategies. No constraints are imposed on the behaviour of agents outside B, but the plausibility operator $\mathbf{Pl}_A$ can be used to change the set of agents (viz A) whose play is restricted. The update/refinement modalities (**set-pl** ω)/(**refn-pl** ω) are used to change the plausibility set Υ_M in the model.

Definition 25 (Semantics of $\mathcal{L}^{\mathbf{base}}_{ATLP}$) *Let $M \in CGSP(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ and $A, B \subseteq \mathbb{A}\mathrm{gt}$. The semantics of ATLP formulae is given as follows:*

$M, q \models_B p$ *iff* $p \in \pi(q)$ *and* $p \in \Pi$

$M, q \models_B \neg\varphi$ *iff* $M, q \not\models_B \varphi$

$M, q \models_B \varphi \wedge \psi$ *iff* $M, q \models_B \varphi$ *and* $M, q \models_B \psi$

$M, q \models_B \langle\langle A\rangle\rangle \bigcirc \varphi$ *iff there is a B-plausible s_A s.t.* $M, \lambda[1] \models_B \varphi$ *for all* $\lambda \in out_M(q, s_A, B)$

$M, q \models_B \langle\langle A\rangle\rangle \Box\varphi$ *iff there is a B-plausible s_A s.t.* $M, \lambda[i] \models_B \varphi$ *for all* $\lambda \in out_M(q, s_A, B)$ *and all* $i \in \mathbb{N}_0$

$M, q \models_B \langle\langle A\rangle\rangle \varphi \mathcal{U} \psi$ *iff there is a B-plausible s_A such that, for all $\lambda \in out_M(q, s_A, B)$, there is $i \in \mathbb{N}_0$ with $M, \lambda[i] \models_B \psi$, and $M, \lambda[j] \models_B \varphi$ for all $0 \leq j < i$*

$M, q \models_B \mathbf{Pl}_A \varphi$ *iff* $M, q \models_A \varphi$

$M, q \models_B$ (**set-pl** ω)φ *iff $M', q \models_B \varphi$ where the new model M' is equal to M but the new set $\Upsilon_{M'}$ of plausible strategy profiles of M' is set to $[\![\omega]\!]^q_M$.*

$M, q \models_B$ (**refn-pl** ω)φ *iff $M', q \models_B \varphi$ where M' is equal to M but $\Upsilon_{M'}$ set to $\Upsilon_M \cap [\![\omega]\!]^q_M$.*

The "absolute" satisfaction relation $\models$ is given by $\models_\emptyset$.

Definition 26 (Validity) *Let $\varphi \in \mathcal{L}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ and $\mathfrak{M} \subseteq CGSP(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$. Formula φ is* valid *with respect to $\mathfrak{M}$ if, and only if, $M, q \models \varphi$ for every $M \in \mathfrak{M}$ and state $q \in St_M$.*

Note that an ordinary concurrent game structure (without plausibility) can be interpreted as a CGSP with all strategy profiles assumed plausible, i.e., with $\Upsilon = \Sigma$, and empty set of plausibility terms Ω.

Let us clarify the semantics behind $\mathbf{Pl}_B\langle\langle A\rangle\rangle\gamma$ once more. The proponents (A) look for a strategy that enforces γ; some of them ($A \cap B$) are assumed to play a part of a plausible strategy profile while the others ($A \setminus B$) can choose an arbitrary collective strategy. Analogously, some opponents ($B \setminus A$) are supposed to play plausibly (that complies to set Υ_M together with the strategy already chosen by $A \cap B$), while the rest ($\mathbb{A}\mathrm{gt} \setminus (A \cup B)$) have unrestricted choice. In particular, when $B = A$, only the choices of the proponents are restricted; for $B = \mathbb{A}\mathrm{gt} \setminus A$ plausibility restrictions apply to the opponents only.

Remark 16 *We observe that our framework is semantically similar to the approach of* social laws *[122, 108, 132]. However, we refer to* strategy profiles *as rational or not, while social laws define constraints on agents'* individual actions. *Also, our motivation is different: In our framework, agents are expected to behave in a specified way because it is rational in some sense; social laws prescribe behaviour sanctioned by social norms and legal regulations.*

Example 26 (Asymmetric matching pennies ctd.) *Suppose that it is plausible to expect that both agents are rational in the sense that they only play undominated strategies.*[6] *Then,* $\Upsilon = \{(s_h, s_h), (s_h, s_t)\}$. *Under this assumption, agent* 2 *is free to grant itself with the prize or to refuse it:* $\mathbf{Pl}\,(\langle\langle 2\rangle\rangle\Diamond\ \mathsf{money}_2 \wedge \langle\langle 2\rangle\rangle\Box\neg\mathsf{money}_2)$. *Still, it cannot choose to win without making the other player win too:* $\mathbf{Pl}\,\neg\langle\langle 2\rangle\rangle\Diamond(\mathsf{money}_2 \wedge \neg\mathsf{money}_1)$. *Likewise, if rationality is defined via iterated undominated strategies, then we have* $\Upsilon = \{(s_h, s_h)\}$, *and therefore the outcome of the game is completely determined:* $\mathbf{Pl}\,\langle\langle\emptyset\rangle\rangle\Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1 \wedge \mathsf{money}_2)$.

Note that, in order to include both *notions of rationality in the model, we can encode them as denotations of two different plausibility terms – say,* ω_{undom} *and* ω_{iter}*, with* $[\![\omega_{undom}]\!]^{q_0} = \{(s_h, s_h), (s_h, s_t)\}$*, and* $[\![\omega_{iter}]\!]^{q_0} = \{(s_h, s_h)\}$. *Let* M_1' *be model* M_1 *with plausibility terms and their denotation defined as above. Then, we have that* $M_1', q_0 \models (\textbf{\textit{set-pl}}\ \omega_{undom})\mathbf{Pl}\,(\langle\langle 2\rangle\rangle\Diamond\mathsf{money}_2 \wedge \langle\langle 2\rangle\rangle\Box\neg\mathsf{money}_2) \wedge (\textbf{\textit{set-pl}}\ \omega_{iter})\mathbf{Pl}\,\langle\langle\emptyset\rangle\rangle\Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1 \wedge \mathsf{money}_2)$.

Out of many solution concepts, Nash equilibrium is the most widely accepted, especially for non-cooperative games. We briefly extend our working example with game analysis based on Nash equilibrium. Note that, in this case, it is not possible to define rationality with independent constraints on agents' individual strategies (like in normative systems). These are full strategy profiles being rational or not, since rationality of a strategy depends on the actual response of the other players.

Example 27 (Asymmetric matching pennies ctd.) *Suppose that rationality is defined through Nash equilibria. Then,* $\Upsilon = \{(s_h, s_h), (s_t, s_t)\}$. *Under this assumption, agent* 2 *is sure to get the prize:* $\mathbf{Pl}\,\langle\langle\emptyset\rangle\rangle\Box(\neg\mathsf{start} \rightarrow \mathsf{money}_2)$.

Moreover, by choosing the right strategy, 2 *can control the outcome of the other agent:* $\mathbf{Pl}\,(\langle\langle 2\rangle\rangle\Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1) \wedge \langle\langle 2\rangle\rangle\Box\neg\mathsf{money}_1)$. *Note that agent* 1 *can control her own outcome too, if we assume that the players are obliged to play rationally:* $\mathbf{Pl}\,(\langle\langle 1\rangle\rangle\Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1) \wedge \langle\langle 1\rangle\rangle\Box\neg\mathsf{money}_1)$. *This may seem strange, but a Nash equilibrium assumes implicitly that the agents coordinate their actions somehow. Then, assuming a particular choice of one agent in advance* constrains *the other agent responses considerably, which puts the first agent at advantage.*

Example 28 (Bargaining ctd.) *Let* ω_{NE} *denote the set of Nash equilibria (every payoff can be reached by a Nash equilibrium), and* ω_{SPN} *the set of subgame perfect Nash equilibria in the game. Then, the following holds for every* $x \in [0, 1]$*:*

$M_2', q_0 \models$

$(\textbf{\textit{set-pl}}\ \omega_{NE})\langle\langle 1, 2\rangle\rangle\Diamond(\mathsf{p}_1^{x} \wedge \mathsf{p}_2^{1-x}) \wedge (\textbf{\textit{set-pl}}\ \omega_{SPN})\langle\langle\emptyset\rangle\rangle\Diamond(\mathsf{p}_1^{\frac{1-\delta_2}{1-\delta_1\delta_2}} \wedge \mathsf{p}_2^{\frac{\delta_2(1-\delta_1)}{1-\delta_1\delta_2}})$.

[6] We recall from Section 5.2.1 that a strategy $s_a \in \Sigma_a$ is called *undominated* if, and only if, there is no strategy $s_a' \in \Sigma_a$ such that the achieved utility of s_a' is at least as good as for s_a for all counterstrategies $s_{-a} \in \Sigma_{\mathbb{A}\mathrm{gt}\setminus\{a\}}$ and strictly better for at least one counterstrategy $s_{-a} \in \Sigma_{\mathbb{A}\mathrm{gt}\setminus\{a\}}$.

where M_2' is given by M_2 extended by plausibility terms and their denotation as introduced above.

Finally, we observe that the "plausibility refinement" operator can be used to combine several solution concepts, e.g., (**set-pl** ω_{NE})(**refn-pl** ω_{PO}) restricts plausible play to *Pareto optimal Nash equilibria*. We can also use (**refn-pl** $\cdot$) to *compare* different notions of rationality. For example, (**set-pl** ω_{NE})(**refn-pl** ω_{PO})$\langle\langle \mathbb{A}\mathrm{gt}\rangle\rangle\bigcirc\top$ can be used to check if Pareto optimal NE's exist in the model at all.

The base language $\mathcal{L}^{\text{base}}_{ATLP}$ allows to restrict the analysis to a subset of available strategy profiles. One drawback of $\mathcal{L}^{\text{base}}_{ATLP}$ is that we cannot specify sets of plausible/rational strategy profiles *in the object language*, simply because our terms do not have any internal structure — they are just constants. Ideally, one would like to have a flexible language of terms that allows to specify *any sensible rationality assumption*, and then impose it on the system.

Our first step is to employ formulae of ATLI and make use of the results in Section 5.3.4. The second step is to define a proper extension of $\mathcal{L}^{\text{base}}_{ATLP}$ where these concepts can be expressed, thus enabling both specification of plausibility and reasoning about plausible behaviour to be conducted in ATLP. The idea is to use ATLP formulae θ to specify sets of plausible strategy profiles, with the intended meaning that Υ collects exactly the profiles for which θ holds. Then, we can embed such an ATLP-based plausibility specification in another formula of ATLP.

5.4.3 Plausibility Terms based on ATLI

Definition 27 ($\mathcal{L}_{ATLP^{ATLI}}$) *Let $\Omega^* = \{(\sigma.\theta) \mid \theta \in \mathcal{L}_{ATLI}(\mathbb{A}\mathrm{gt}, \Pi, \{\sigma[1], \ldots, \sigma[k]\})\}$. That is, Ω^* collects terms of the form $(\sigma.\theta)$, where θ is an ATLI formula including only references to individual agents' parts of the strategy profile σ.[7] The language of $ATLP^{ATLI}$ is defined as $\mathcal{L}^{base}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega^*)$.*

The idea behind terms of this form is simple. We have an ATLI formula θ, parameterized with a variable σ that ranges over the set of strategy profiles Σ. Now, we want $(\sigma.\theta)$ to denote exactly the set of profiles from Σ, for which formula θ holds. However – as σ denotes a strategy profile, and ATLI allows only to refer to strategies of individual agents – we need a way of addressing substrategies of σ in θ. This can be done by using ATLI terms $\sigma[i]$, which are interpreted as i's substrategy in σ.

For example, we may assume that a rational agent does not grant the other agents with too much control over its life: $(\sigma\ .\ \bigwedge_{a\in\mathbb{A}\mathrm{gt}}((\mathbf{str}_a\,\sigma[a])$ $\neg\langle\langle \mathbb{A}\mathrm{gt}\setminus\{a\}\rangle\rangle\Diamond \mathsf{dead_a}))$. Note that games defined by CGS's are, in general, not determined, so the above specification does not guarantee that each rational agent can efficiently protect her life. It only requires that she should behave cautiously so that her opponents do not have complete power to kill her.

Definition 28 (Denotation of ATLI-based plausibility terms) *Let M be a CGS of the form $M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o\rangle$ and Ω^* be as in Definition 27.*

[7] σ is the only variable in θ and refers to a strategy profile.

For each $s \in \Sigma$ we define M^s to be the following CGS with intentions:

$$M^s = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o, \mathcal{I}^0, \mathfrak{Str}, [\cdot] \rangle$$

with $\mathfrak{Str}_a = \{\sigma[a]\}$, and $[\sigma[a]] = s[a]$. We recall from Section 5.3.4 that $\mathcal{I}^0$ represents the full intention relation.

The plausibility mapping for terms from Ω^* is defined as:

$$[\![\sigma.\theta]\!]^q = \{s \in \Sigma \mid M^s, q \models \theta\}.$$

It is now possible to plug in arbitrary ATLI specifications of rationality, and reason about their consequences.

Example 29 (Asymmetric matching pennies ctd.) *It seems that explicit quantification over the opponents' responses (not available in ATLI) is essential to express undominatedness of strategies (cf. [130] and Section 5.5.3). Still, we can at least assume that a rational player should avoid playing strategies that* guarantee *failure if a potentially successful strategy is available. Under this assumption, player 1 should never play tail, and in consequence player 2 controls the outcome of the game:*

$$M_1'', q_0 \models (\textbf{\textit{set-pl}}\ \sigma.\ \textstyle\bigwedge_{a \in \mathbb{A}\mathrm{gt}} (\langle\!\langle \mathbb{A}\mathrm{gt} \rangle\!\rangle \Diamond \mathsf{money_a} \rightarrow (\mathbf{str}_a\, \sigma[a]) \langle\!\langle \mathbb{A}\mathrm{gt} \rangle\!\rangle \Diamond \mathsf{money_a})) \\ \mathbf{Pl}\,\big(\langle\!\langle 2 \rangle\!\rangle \Diamond (\mathsf{money_1} \wedge \mathsf{money_2}) \wedge \langle\!\langle 2 \rangle\!\rangle \Box \neg (\mathsf{money_1} \wedge \mathsf{money_2})\big).$$

where M_1'' is the CGS M_1 extended with propositions $\mathsf{p}_\mathsf{i}^1 \equiv \mathsf{money_i}$, ATLI-based plausibility terms, and their denotation according to Definition 28.

Moreover, if only Pareto optimal strategy profiles can be played, then both players are bound to keep winning money:

$$M_1'', q_0 \models (\textbf{\textit{set-pl}}\ \sigma.PO^{\Diamond}(\sigma))\ \mathbf{Pl}\, \langle\!\langle \emptyset \rangle\!\rangle \Box (\neg \mathsf{start} \rightarrow \mathsf{money_1} \wedge \mathsf{money_2}).$$

Finally, restricting plausible strategy profiles to Nash equilibria guarantees that player 2 should plausibly get money, but the outcome of player 1 is not determined:

$$M_1'', q_0 \models (\textbf{\textit{set-pl}}\ \sigma.NE^{\Diamond}(\sigma))\ \mathbf{Pl}\,\big(\langle\!\langle \emptyset \rangle\!\rangle \Box (\neg \mathsf{start} \rightarrow \mathsf{money_2}) \\ \wedge \neg \langle\!\langle \emptyset \rangle\!\rangle \Diamond \mathsf{money_1} \wedge \neg \langle\!\langle \emptyset \rangle\!\rangle \Box \neg \mathsf{money_1}\big).$$

Example 30 (Bargaining ctd.) *For the bargaining agents and $\kappa = (1 - \delta_2)\frac{1-(\delta_1\delta_2)^{\frac{T}{2}}}{1-\delta_1\delta_2} + (\delta_1\delta_2)^{\frac{T}{2}}$, we have accordingly:*

1. $M_2', q_0 \models (\textbf{\textit{set-pl}}\ \sigma.NE^{\Diamond}(\sigma))\mathbf{Pl}\, \langle\!\langle \emptyset \rangle\!\rangle \bigcirc (\mathsf{p}_1^{\mathsf{x}} \wedge \mathsf{p}_2^{1-\mathsf{x}})$ *for every x;*
2. $M_2', q_0 \models (\textbf{\textit{set-pl}}\ \sigma.SPN^{\Diamond}(\sigma))\mathbf{Pl}\, \langle\!\langle \emptyset \rangle\!\rangle \bigcirc (\mathsf{p}_1^{\kappa} \wedge \mathsf{p}_2^{1-\kappa})$*;*
3. $M_2', q_0 \models (\textbf{\textit{set-pl}}\ \sigma.SPN^{\Diamond}(\sigma))\mathbf{Pl}\, \langle\!\langle \emptyset \rangle\!\rangle \Box (\neg \mathsf{p}_1^{\mathsf{x}_1} \wedge \neg \mathsf{p}_2^{\mathsf{x}_2})$ *for every $x_1 \neq \kappa$ and $x_2 \neq 1 - \kappa$*

where M_2' is the CGSP obtained from CGS M_2 by adding ATLI-based plausibility terms and their denotation.

Thus, we can encode a game as a CGS M, specify rationality assumptions with an ATLI formula θ, and ask if a desired property φ of the system holds under these assumptions by model checking (**set-pl** $\sigma.\theta)\varphi$ in M. Note that the denotation of plausibility terms in Ω^* is fixed. We report our results on the complexity of solving such games in Section 5.6.

5.4.4 Language $\mathcal{L}^k_{ATLP}$ and $\mathcal{L}^\infty_{ATLP}$

As we have already explained, our main idea is to use ATLP for both specification of rationality assumptions and describtion of the outcome of rational play. Thus, we need a possibility to embed an ATLP formula φ (that defines the rationality condition) in a "higher-level" formula of ATLP, as a part of plausibility term (**set-pl** $\sigma.\varphi$). The reading of (**set-pl** $\sigma.\varphi)\psi$ is, again: "Let the plausibility set consist of profiles σ that satisfy φ; then, ψ holds". Apart from the possibility of nesting formulae via plausibility updates, we also propose to add quantifier-like structures to the language of terms. Consider, for example, the term $\sigma_1.(\exists\sigma_2)\varphi$. We would like to *collect* all strategies s_1 such that there is a strategy s_2 for which φ holds (we use σ_i to refer to s_i). Thus, $\sigma_1.(\exists\sigma_2)\varphi$ is supposed to act in a similar way as the first order logic-based set specification $\{x \mid \exists y : \varphi(x,y)\}$. It is easy to see that e.g. the set of all undominated strategies can now be specified in a straightforward way.

As before, the new version of ATLP is given over a set $\mathbb{A}\mathrm{gt} = \{a_1,\ldots,a_k\}$ of agents, a set Π of propositions, and a set Ω of *primitive plausibility terms* (cf. Section 5.3.4). In addition to these sets, we also include a set $\mathcal{V}ar$ of *strategic variables*. Variables in $\mathcal{V}ar$ range over strategy profiles; we need them to characterize specific rationality criteria, in a way similar to first order logic specifications.

The definition of $\mathcal{L}_{ATLP}$ is given recursively. In each step the structure of plausibility terms becomes more sophisticated. At first, we only consider terms out of Ω; their interpretation is given in the model. On the next level, we also allow plausibility terms to be quantified ATLP formulae which contain strategic variables *and* elements from Ω. Plausibility terms of subsequent levels can again be based on terms from the previous levels, and so forth. In consequence, the *core 0-level language* of our new ATLP is almost the same as the base language $\mathcal{L}^{\text{base}}_{ATLP}$ defined in Section 5.4.1: It extends it with simple combinations of terms.

In general, all the levels of the language can be seen as containing ordinary formulae of the original ATLP, the only thing that changes as we move to higher levels is the complexity of plausibility terms. We begin with defining simple combinations of plausibility terms, and then present the hierarchy of languages $\mathcal{L}^k_{ATLP}$, with the underlying idea that $\mathcal{L}^k_{ATLP}$ allows for at most k ($k \in \mathbb{N}_0$) nested plausibility updates. The full language $\mathcal{L}_{ATLP}$ allows for any arbitrary finite number of nestings.

Definition 29 (Strategic combination) *Let* $\mathbb{A}\mathrm{gt}$ *denote a set of agents and* X *be a non-empty set of symbols. We say that* y *is a* strategic combination *of* x *if it is generated by the following grammar:*

$$y ::= x \mid \langle y,\ldots,y\rangle \mid y[A]$$

where $x \in X$, $\langle y,\ldots,y\rangle$ *is a vector of length* $|\mathbb{A}\mathrm{gt}|$*, and* $A \subseteq \mathbb{A}\mathrm{gt}$*. The set of* strategic combinations *over* X *is defined by* $\mathcal{T}(X)$*. It is easy to see that operator* $\mathcal{T}$ *is idempotent (*$\mathcal{T}(X) = \mathcal{T}(\mathcal{T}(X))$*).*

The intuition is that elements of $x \in X$ are symbols in the object language that refer to sets of strategy profiles, and the elements of $\mathcal{T}(X)$ allow to

combine these sets to new sets.[8] Let x refer to a set of strategy profiles $\chi \subseteq \Sigma$. Then, $x[A]$ refers to all the profiles in Σ in which A's substrategy agrees with some profile from χ. Similarly, if $x_1, \ldots, x_k$ denote sets of strategy profiles $\chi_1, \ldots, \chi_k$, then $\langle x_1, \ldots, x_k \rangle$ refers to all the profiles that agree on a_i's strategy with at least one profile from χ_i for each $i = 1, \ldots, k$.

Definition 30 ($\mathcal{L}^k_{ATLP}$) *Let $\mathbb{A}\mathrm{gt}$ be a set of agents, Π a set of propositions, Ω be a set of primitive plausibility terms, and $\mathcal{V}ar$ a set of* strategic variables *(with typical element σ). The logics $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \mathcal{V}ar, \Omega)$, $k = 0, 1, 2, \ldots$, are recursively defined as follows:*

- $\mathcal{L}^0_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \mathcal{V}ar, \Omega) = \mathcal{L}^{base}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega_0)$, *where* $\Omega_0 = \mathcal{T}(\Omega)$;
- $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \mathcal{V}ar, \Omega) = \mathcal{L}^{base}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega_k)$, *where:*
$$\Omega_k = \mathcal{T}(\Omega_{k-1} \cup \Omega^k),$$
$$\Omega^k = \{\sigma_1.(Q_2\sigma_2)\ldots(Q_n\sigma_n)\varphi \mid n \in \mathbb{N}, \forall i\, (1 \leq i \leq n \Rightarrow \sigma_i \in \mathcal{V}ar,\ Q_i \in \{\forall, \exists\}, \varphi \in \mathcal{L}^{base}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \mathcal{T}(\Omega_{k-1} \cup \{\sigma_1, \ldots, \sigma_n\})))\}.$$

Thus, plausibility terms on level k (i.e., Ω_k) augment terms from the previous level (Ω_{k-1}) with new terms Ω^k that combine *quantification over strategic variables $\sigma_1, \ldots \sigma_n$ with formulae possibly containing these strategic variables*. Such terms are used to *collect* (or describe) specific strategy profiles (referred to by variable σ_1 which plays a distinctive role in comparison with the other variables).

Definition 31 ($\mathcal{L}_{ATLP}$) *The set of ATLP formulae with arbitrary finite nesting of plausibility terms is defined by*

$$\mathcal{L}_{ATLP} = \mathcal{L}^{\infty}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \mathcal{V}ar, \Omega) = \lim_{k \to \infty} \mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \mathcal{V}ar, \Omega).$$

Definition 32 (k-formula, k-term) *Formula $\varphi \in \mathcal{L}^{\infty}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \mathcal{V}ar, \Omega)$ is called an $ATLP^k$-formula (or simply* k-formula*) if, and only if, $\varphi \in \mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \mathcal{V}ar, \Omega)$. Analogously, a plausibility term occurring in a k-formula is called a* k-term.

Remark 17 *We use the acronym ATLP to refer to both the full language $\mathcal{L}^{\infty}_{ATLP}$ and the basic sublanguage $\mathcal{L}^{base}_{ATLP}$.*

Example 31 (Illustrating plausibility terms in $\mathcal{L}^k_{ATLP}$) *Below we present some simple formulae illustrating the different levels of our logic.*

$\mathcal{L}^{base}_{ATLP}$: ($\langle\langle$**set-pl** $\omega_{NE}\rangle\rangle$**Pl** $\langle\langle A\rangle\rangle\gamma$*; group A can enforce γ if only Nash equilibria are played (we assume that ω_{NE} denotes exactly the set of Nash equilibria in the model).*

$\mathcal{L}^0_{ATLP}$: ($\langle\langle$**set-pl** $\langle\omega_{NE}, \ldots, \omega_{NE}\rangle\rangle$**Pl** $\langle\langle A\rangle\rangle\gamma$*; plausibility terms can be combined. Note the difference to the previous formula, agents are assumed to play a strategy which is part of* some *NE. The resulting strategy profile does not have to be a Nash equilibrium, though.*

[8] This correspondence will be given formally in Definition 33 (Section 5.4.5).

$\mathcal{L}^1_{ATLP}$**:** $\varphi \equiv (\textbf{\textit{set-pl}}\ \sigma.\exists\sigma_1\varphi'(\sigma,\sigma_1))\mathbf{Pl}\langle\langle A\rangle\rangle\gamma$ *where* $\varphi'(\sigma,\sigma_1)$ *is a formula possibly containing operators* $(\textbf{\textit{set-pl}}\ \omega)$ *with* $\omega \in \mathcal{T}(\Omega \cup \{\sigma,\sigma_1\})$*; e.g.* $\varphi' \equiv (\textbf{\textit{set-pl}}\ \langle\sigma,\dots\sigma,\sigma_1,\omega_{NE}\rangle)\mathbf{Pl}\langle\langle A\rangle\rangle\gamma'$*. We will have a closer look at the* $(\textbf{\textit{set-pl}}\ \cdot)$ *operator in* φ*. The operator collects all strategies* σ *such that there exists another strategy profile* σ_1 *for which* $\mathbf{Pl}\langle\langle A\rangle\rangle\gamma'$ *holds if all but the last* 2 *agents play according to* σ*, the second to last agent plays according to* σ_1*, and the last one according to a fixed strategy out of* ω_{NE}*.*

$\mathcal{L}^2_{ATLP}$**:** *Consider the previous formula* φ *again, but this time* $\varphi'(\sigma,\sigma_1)$ *can also contain quantification; e.g.* $\varphi' \equiv ((\textbf{\textit{set-pl}}\ \langle\sigma,\dots,\sigma,\sigma_1,\omega_{NE}\rangle)\mathbf{Pl}\langle\langle B\rangle\rangle\gamma') \rightarrow ((\textbf{\textit{set-pl}}\ \sigma'.\exists\sigma'_1\varphi''(\sigma',\sigma'_1))\mathbf{Pl}\langle\langle A\rangle\rangle\gamma)$ *where* $\varphi''(\sigma',\sigma'_1)$ *is a base formula with plausibility terms taken from* $\mathcal{T}(\Omega \cup \{\sigma',\sigma'_1\})$*.*

In the next section we show how the denotation of complex terms is constructed, and how it is plugged into the semantics of ATLP from Section 5.4.2.

5.4.5 Semantics of $\mathcal{L}^k_{ATLP}$ and $\mathcal{L}^\infty_{ATLP}$

$\mathcal{L}^k_{ATLP}$ does not change the very structure of ATLP formulae, it only extends $\mathcal{L}^{\text{base}}_{ATLP}$ by more ornate plausibility terms. Therefore, it seems natural that the plausibility mapping for theses terms is of particular interest; the denotation reflects the construction of strategic combinations given in Definition 29.

Definition 33 (Extended plausibility mapping $\widehat{[\![\cdot]\!]}$) *The* extended plausibility mapping $\widehat{[\![\cdot]\!]}_M$ *with respect to* $M \in CGSP(\mathbb{A}\text{gt},\Pi,\Omega)$ *is defined as follows:*

1. *If* $\omega \in \Omega$ *then* $\widehat{[\![\omega]\!]}^q_M = [\![\omega]\!]^q_M$*;*
2. *If* $\omega = \omega'[A]$ *then* $\widehat{[\![\omega]\!]}^q_M = \{s \in \Sigma \mid \exists s' \in \widehat{[\![\omega']\!]}^q_M\ s|_A = s'|_A\}$*;*
3. *If* $\omega = \langle\omega_1,\dots\omega_k\rangle$ *then* $\widehat{[\![\omega]\!]}^q_M = \{s \in \Sigma \mid \exists t_1 \in \widehat{[\![\omega_1]\!]}^q_M,\dots,\exists t_k \in \widehat{[\![\omega_k]\!]}^q_M \forall i = 1,...,k\ s|_{a_i} = t_i|_{a_i})\}$*;*
4. *If* $\omega = \sigma_1.(Q_2\sigma_2)\dots(Q_n\sigma_n)\varphi$ *then*

$$\widehat{[\![\omega]\!]}^q_M = \{s_1 \in \Sigma \mid Q_2 s_2 \in \Sigma,\dots,Q_n s_n \in \Sigma \quad (M^{s_1,\dots,s_n}, q \models \varphi)\},$$

where $M^{s_1,\dots,s_n}$ *is equal to* M *except that we fix* $\Upsilon_{M^{s_1,\dots,s_n}} = \Sigma$*,* $\Omega_{M^{s_1,\dots,s_n}} = \Omega_M \cup \{\sigma_1,\dots,\sigma_n\}$*,* $[\![\sigma_i]\!]^q_{M^{s_1,\dots,s_n}} = \{s_i\}$*, and* $[\![\omega]\!]^q_{M^{s_1,\dots,s_n}} = [\![\omega]\!]^q_M$ *for all* $\omega \neq \sigma_i$*,* $1 \leq i \leq n$*, and* $q \in St_M$*. That is, the denotation of* σ_i *in* $M^{s_1,\dots,s_n}$ *is set to strategy profile* s_i*.*[9]

Consider, for instance, plausibility term $\sigma_1.\forall\sigma_2\varphi$. The extended plausibility mapping $\widehat{[\![\sigma_1.\forall\sigma_2\varphi]\!]}_q$ collects all strategy profiles $s_1 \in \Sigma$ (referred to by σ_1) such that *for all* strategy profiles $s_2 \in \Sigma$ (referred to by σ_2) φ is true in model M^{s_1,s_2} and state $q \in St$, i.e. $M^{s_1,s_2}, q \models \varphi$ for all $s_2 \in \Sigma$.

[9] It should be emphasized that model $M^{s_1,\dots,s_n}$ in which plausibility of profile s_1 is evaluated does *not* presuppose any notion of plausibility, i.e., $\Upsilon_{M^{s_1,\dots,s_n}} = \Sigma$.

Remark 18 *Note that if the language includes a term $\omega_\top$ that refers to* all *strategy profiles, then $x[A]$ can be expressed as $\langle \omega_1, \ldots, \omega_k \rangle$, where $\omega_a = x_a$ for $a \in A$, and $\omega_a = \omega_\top$ otherwise. We also observe that in $\mathcal{L}^k_{ATLP}$, $k > 0$, $\omega_\top$ can be expressed as $\sigma.\top$.*

In Definition 25 we defined the semantics of the base language of ATLP. Truth of $\mathcal{L}^k_{ATLP}$ formulae is defined in the same way, we only need to replace the previous (simple) plausibility mapping by the extended one in the semantics of plausibility updates.

Definition 34 (Semantics of $\mathcal{L}^k_{ATLP}$ and $\mathcal{L}^\infty_{ATLP}$) *The semantics for $\mathcal{L}_{ATLP}$ formulae is given as in Definition 25 with the extended plausibility mapping $\widehat{[\![\cdot]\!]}_M$ used instead of $[\![\cdot]\!]_M$. I.e., only the semantic clauses for $(\textbf{\textit{set-pl}}\ \omega)\varphi$ and $(\textbf{\textit{refn-pl}}\ \omega)\varphi$ change as follows:*

$M, q \models_B (\textbf{set-pl}\ \omega)\varphi$ *iff* $M', q \models_B \varphi$ *where the new model M' is equal to M but the new set $\Upsilon_{M'}$ of plausible strategy profiles is set to $\widehat{[\![\omega]\!]}^q_M$;*

$M, q \models_B (\textbf{refn-pl}\ \omega)\varphi$ *iff* $M', q \models_B \varphi$ *where the new model M' is equal to M but the new set $\Upsilon_{M'}$ of plausible strategy profiles is set to $\Upsilon_M \cap \widehat{[\![\omega]\!]}^q_M$.*

Remark 19 *By a slight abuse of notation, we will refer to the extended plausibility mapping with the same symbol as to the simple plausibility mapping, i.e., with $[\![\cdot]\!]$.*

We will discuss some important examples of $\mathcal{L}_{ATLP}$ formulae and terms (together with their interpretation) in Sections 5.5.2 and 5.5.3 where ATLP characterizations of solution concepts are presented.

5.5 Properties of ATLP

This section contains our main conceptual results. We show:

1. That several logics can be embedded into ATLP by means of polynomial translation of models and/or formulae (Section 5.5.1),
2. That several classical solution concepts for extensive games (Nash equilibria, subgame perfect Nash equilibria, Pareto Optimality), can be characterized in ATLP already in the language $\mathcal{L}^1_{ATLP}$ (Section 5.5.2),
3. That these solution concepts can be also re-formulated in a qualitative way, through appropriate formulae of ATLP parameterized by ATL path formulae (Section 5.5.3).

5.5.1 Embedding Existing Logics into ATLP

In this section, we compare ATLP with several related logics and show their formal relationships. To this end, we first define notions that allow to compare expressivity of logical systems. *Embedding* takes place on the level of satisfaction relations ($\models$): Logic L_1 embeds L_2 if models and formulae of L_2 can be simulated in L_1 in a truth-preserving way. *Subsumption* refers to the

level of valid sentences: L_1 subsumes L_2 if all the validities of L_2 are validities of L_1 as well.

Definition 35 (Embedding) Logic L_1 embeds logic L_2 *iff there is a translation* tr *of* L_2 *formulae into formulae of* L_1*, and a transformation* TR *of* L_2 *models into models of* L_1*, such that* $M, q \models_{L_2} \varphi$ *iff* $TR(M), q \models_{L_1} tr(\varphi)$ *for every pointed model* M, q *and formula* φ *of* L_2*.*

Note that the translation of formulae and transformation of models are supposed to be independent. This prevents translation schemes that transform triples $M, q \models \varphi$ in L_2 to $M', q \models \top$, and triples $M, q \not\models \varphi$ in L_2 to $M', q \not\models \bot$ (with an arbitrary model M'), that would yield embeddings between any pair of logics.

It is important to point out that all the transformation and translation schemes proposed in this section can be computed in polynomial time and incur only polynomial increase in the size of models and the length of formulae. Thus, we are in fact interested in *polynomial* embeddings of logics in ATLP.

Definition 36 (Subsumption) Logic L_1 subsumes logic L_2 *iff the set of validities of* L_1 *subsumes validities of* L_2*.*

Proposition 48 *ATLP embeds ATL.*

Proof We use the identity translation of formulae: $tr(\varphi) \equiv \varphi$. As for models, $TR(M) = M'$ that extends M with an arbitrary set of plausible strategy profiles Υ. It is easy to see that the plausibility assumptions Υ will never be used in the evaluation of φ since φ includes no **Pl** operators. Thus, the result of the evaluation will be the same as for $M, q \models \varphi$. ■

The above reasoning implies also that ATL validities hold for all ATLP models.

Corollary 5 *ATLP subsumes ATL.*

The relationship of ATLP to most other logics can be studied only in the context of embedding, as they use different modal operators (and thus yield incomparable sets of valid formulae). We begin with embedding "ATL with Intentions" [85] in ATLP. Then, we show that "CTL with Plausibility" from [25] can be embedded in ATLP for a limited (but very natural) class of models. Finally, we propose an embedding of the two existing versions of Game Logic with Preferences [137, 138] which allow to reason about what can happen under *particular* game-theoretical rationality assumptions.

Proposition 49 *ATLP embeds ATLI.*

Proof sketch For an ATLI model $M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o, \mathcal{I}, \mathfrak{Str}, [\cdot] \rangle$, we construct the corresponding concurrent game structure with plausibility $TR(M) = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o, \Upsilon, \Omega, [\![\cdot]\!] \rangle$ with the set of plausible profiles $\Upsilon = \{s \in \Sigma \mid s \text{ is consistent with } \mathcal{I}\}$, plausibility terms $\Omega = \{\omega_\sigma \mid \sigma \in \mathfrak{Str}\} \cup \{\omega_\top\}$, and their denotation $[\![\omega_\top]\!]^q = \Sigma$ and $[\![\omega_\sigma]\!]^q = \{s \in \Sigma \mid s|_a = [\sigma]\}$ for each $\sigma \in \mathfrak{Str}_a$.

For an ATLI formula φ, we construct its ATLP translation by transforming strategic assumptions (about agents' intentions) imposed by $(\mathbf{str}_a\,\sigma)$ to plausibility assumptions (about strategy profiles that can be plausibly played) defined by (**set-pl** ω_σ) and applying them to the appropriate set of agents (i.e., those for whom strategic assumptions have been defined). Formally, the translation is defined as $tr(\varphi) = \mathbf{Pl}\, tr_{\langle\omega_\top,\ldots,\omega_\top\rangle}(\varphi)$, where $tr_{\langle\omega_1,\ldots,\omega_k\rangle}$ is defined as follows:

$$\begin{array}{rcl}
tr_{\langle\omega_1,\ldots,\omega_k\rangle}(p) & = & p,\\
tr_{\langle\omega_1,\ldots,\omega_k\rangle}(\neg\varphi) & = & \neg tr_{\langle\omega_1,\ldots,\omega_k\rangle}(\varphi),\\
tr_{\langle\omega_1,\ldots,\omega_k\rangle}(\varphi_1\wedge\varphi_2) & = & tr_{\langle\omega_1,\ldots,\omega_k\rangle}(\varphi_1)\wedge tr_{\langle\omega_1,\ldots,\omega_k\rangle}(\varphi_2),\\
tr_{\langle\omega_1,\ldots,\omega_k\rangle}(\langle\!\langle A\rangle\!\rangle\bigcirc\varphi) & = & \langle\!\langle A\rangle\!\rangle\bigcirc tr_{\langle\omega_1,\ldots,\omega_k\rangle}(\varphi),\\
 & & \text{(for } \langle\!\langle A\rangle\!\rangle\Box\varphi \text{ and } \langle\!\langle A\rangle\!\rangle\varphi_1\,\mathcal{U}\,\varphi_2 \text{ analogously)}\\
tr_{\langle\omega_1,\ldots,\omega_k\rangle}((\mathbf{str}_a\,\sigma'_a)\varphi) & = & (\textbf{set-pl}\ \overrightarrow{\omega})tr_{\overrightarrow{\omega}}(\varphi),\\
\text{where}\quad \overrightarrow{\omega} & = & \langle\omega_1,\ldots,\omega_{\sigma'_a},\ldots,\omega_k\rangle.
\end{array}$$

Note that, for "vanilla" ATLI, $\langle\!\langle A\rangle\!\rangle\gamma$ holds iff γ can be enforced *against every response strategy from* $\mathbb{A}\mathrm{gt}\setminus A$. Thus, e.g., $M,q\models_{\mathrm{ATLI}}(\mathbf{str}_a\,\sigma_a)\langle\!\langle A\rangle\!\rangle\Box p$ iff $TR(M),q\models_{\mathrm{ATLP}}\mathbf{Pl}\,(\textbf{set-pl}\ \langle\omega_\top,\ldots,\omega_{\sigma_a},\ldots,\omega_\top\rangle)\langle\!\langle A\rangle\!\rangle\Box p$, and analogously for the other cases. ■

CTLP, i.e., "CTL with Plausibility" [25], is an extension of the branching-time logic CTL with a similar notion of plausibility as the one we use here. The main difference lies in the fact that CTLP formulae refer to plausible *paths* rather than strategy profiles.

Proposition 50 *ATLP embeds CTLP in the class of transition systems.*

Proof sketch To transform models, we first observe that every transition system M can be seen as a concurrent game structure that includes only a single agent a_1. Furthermore, we can transform M to a CGSP $TR(M)$ by adding $\Upsilon=\Sigma$ and $\Omega=\emptyset$ (cf. Section 5.4.1). To translate CTLP formulae, we use the scheme below:

$$\begin{array}{rcl}
tr(\varphi) & = & tr_{(\textbf{set-pl}\ \sigma.\top)}(\varphi),\\
tr_\omega(p) & = & p,\\
tr_\omega(\neg\varphi) & = & \neg tr_\omega(\varphi),\\
tr_\omega(\varphi_1\wedge\varphi_2) & = & tr_\omega(\varphi_1)\wedge tr_\omega(\varphi_2),\\
tr_\omega(\mathsf{E}\gamma) & = & \langle\!\langle\mathbb{A}\mathrm{gt}\rangle\!\rangle tr_\omega(\gamma),\\
tr_\omega(\bigcirc\varphi) & = & \bigcirc tr_\omega(\varphi)\quad \text{(for } \Box\varphi \text{ and } \varphi_1\,\mathcal{U}\,\varphi_2 \text{ analogously);}\\
tr_\omega(\mathbf{Pl}\,\varphi) & = & (\textbf{set-pl}\ \omega)\mathbf{Pl}\,tr_\omega(\varphi),\\
tr_\omega(\mathbf{Ph}\,\varphi) & = & \mathbf{Ph}\,tr_\omega(\varphi),\\
tr_\omega((\textbf{set-pl}\ \gamma)\varphi) & = & tr_{\omega'}(\varphi),\\
\text{where}\quad \omega' & = & \sigma.(\textbf{set-pl}\ \sigma)\mathbf{Pl}\,\langle\!\langle\emptyset\rangle\!\rangle\gamma.
\end{array}$$

Now, $M,q\models_{\mathrm{CTLP}}\varphi$ iff $M,q\models_{\mathrm{ATLP}}tr(\varphi)$.

Note that we cannot use the above construction for arbitrary models of CTLP, as not every set of (plausible) paths can be obtained by memoryless strategy profiles. ■

Proposition 51 *ATLP cannot be polynomially embedded in neither ATL, nor ATLI, nor CTLP.*

Proof Suppose that any of these logics polynomially embeds ATLP. Then, the embedding provides a polynomial reduction of model checking from ATLP to that logic. Since model checking of ATL, ATLI, and CTLP can be done in polynomial deterministic time [8, 85, 25], we get that the problem for ATLP is in $\mathbf{P}$, too. But model checking ATLP is $\mathbf{\Delta_3^P}$-hard already for $\mathcal{L}^{\text{base}}_{ATLP}$ (see Section 5.6). ■

There is not much work on logical descriptions of behaviour of agents under rationality assumptions based on game-theoretical solution concepts. In fact, we know only of one such logic for agents with perfect information, which is GLP from [138]. There, agents can be assumed qualitative preferences (i.e., a propositional formula φ_0 that they supposedly want to make eventually true). Moreover, they are assumed to play rationally in the sense that if they have some strategies that guarantee $\Diamond\varphi_0$, they can use only those strategies in their play. Interestingly enough, the preference criterion was different in a preliminary version of GLP [137], where it was based on the notion of Nash equilibrium. Both versions of GLP can be embedded in ATLP. One may embed game logics with other preference criteria in an analogous way.

Proposition 52 *GLP can be embedded in ATLP.*

Proof sketch For the translation of models, we transform game trees of GLP to concurrent game structures using the construction from Section 5.3.3, and transform the CGS to CGSP by taking $\Upsilon = \Sigma$ and $\Omega = \emptyset$. Then, we use the following translation of GLP formulae:

$$\begin{array}{rcl}
tr(\varphi) & = & \mathbf{Pl}\ tr_{(\textbf{set-pl}\ \sigma.\top)}(\varphi), \\
tr_\omega(p) & = & p, \quad tr_\omega(\neg\varphi) = \neg tr_\omega(\varphi), \quad tr_\omega(\varphi\vee\psi) = tr_\omega(\varphi)\vee tr_\omega(\psi), \\
tr_\omega(\Box\varphi_0) & = & \langle\!\langle\emptyset\rangle\!\rangle\Diamond\varphi_0, \\
tr_\omega([a:\varphi_0]\psi) & = & (\textbf{set-pl}\ \omega')tr_{\omega'}(\psi), \\
\text{where}\quad \omega' & = & \sigma.\mathbf{Pl}\,(\textbf{set-pl}\ \omega)\langle\!\langle\emptyset\rangle\!\rangle\Box\big(plausible(\sigma)\wedge prefers(a,\sigma,\varphi_0)\big) \\
plausible(\sigma) & \equiv & (\textbf{refn-pl}\ \sigma)\langle\!\langle\mathbb{A}\mathrm{gt}\rangle\!\rangle\bigcirc\top \\
prefers(a,\sigma,\varphi_0) & \equiv & \langle\!\langle a\rangle\!\rangle\Diamond\varphi_0\rightarrow(\textbf{refn-pl}\ \sigma[a])\langle\!\langle\emptyset\rangle\!\rangle\Diamond\varphi_0.
\end{array}$$

That is, with each subsequent preference operator $[a:\varphi_0]$, only those from the (currently) plausible strategy profiles are selected that are preferred by a. The preference is based on the (subgame perfect) enforceability of the outcome φ_0 at the end of the game: if φ_0 can be enforced at all, then a prefers strategies that do enforce it.

Now, we have that $\Gamma\models_{\text{GLP}}\varphi$ iff $TR(\Gamma),\emptyset\models_{\text{ATLP}} tr(\varphi)$.[10] ■

Proposition 53 *Preliminary GLP can be embedded in ATLP.*

[10] Again, $\emptyset$ denotes the position with empty history, i.e., the initial state of the game.

Proof Analogous to Proposition 52. The translation only differs in the characterization of agents' preferences. The agents are now assumed to stick to their individual parts of Nash equilibria defined by a zero-sum game where a wins iff φ_0 is enforced:[11]

$$\begin{array}{rcl}
\omega' & = & \sigma.\exists\sigma' \mathbf{Pl}\,(\textbf{set-pl}\ \omega)\langle\langle\emptyset\rangle\rangle\Box \\
 & & \big(plausible(\sigma) \wedge NE(\sigma', a, \varphi_0) \wedge coincides(\sigma, \sigma', a)\big) \\
plausible(\sigma) & \equiv & (\textbf{refn-pl}\ \sigma)\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\bigcirc\top \\
coincides(\sigma, \sigma', a) & \equiv & (\textbf{set-pl}\ \sigma[a])(\textbf{refn-pl}\ \sigma'[a])\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\bigcirc\top \\
NE(\sigma, a, \varphi_0) & \equiv & \bigwedge\limits_{i\in\mathbb{A}\mathrm{gt}} BR_i(\sigma, a, \varphi_0), \\
BR_i(\sigma, a, \varphi_0) & \equiv & \begin{cases}
(\textbf{refn-pl}\ \sigma[\mathbb{A}\mathrm{gt}\setminus\{i\}])\langle\langle i\rangle\rangle\Diamond\varphi_0 & \\
\qquad \rightarrow (\textbf{refn-pl}\ \sigma)\langle\langle\emptyset\rangle\rangle\Diamond\varphi_0 & \text{for } i = a \\
(\textbf{refn-pl}\ \sigma[\mathbb{A}\mathrm{gt}\setminus\{i\}])\langle\langle i\rangle\rangle\Box\neg\varphi_0 & \\
\qquad \rightarrow (\textbf{refn-pl}\ \sigma)\langle\langle\emptyset\rangle\rangle\Box\neg\varphi_0 & i \neq a
\end{cases}
\end{array}$$

■

A couple other logics were defined for various solution concepts with respect to incomplete information games [136, 135]. We do not study them here, since our framework lacks the notions of knowledge and uncertainty – but it seems a promising area of future research.

Remark 20 *We have presented embeddings of several quite different logics into ATLP, which suggests substantial gain in expressive power. Most of them (ATL, ATLI, and CTLP) are embedded already in the lowest levels of the ATLP hierarchy (i.e., $\mathcal{L}^{base}_{ATLP}$ or $\mathcal{L}^{1}_{ATLP}$ with no quantifiers). GLP formulae with at most k preference operators are embedded in $\mathcal{L}^{k}_{ATLP}$, which is inevitable given their semantics that combines model update and irrevocable strategic quantification (cf. the discussion and the complexity results in [2, 21]).*

5.5.2 Classical Solution Concepts in $\mathcal{L}^{1}_{ATLP}$

In Section 5.3.3 we showed how extensive games Γ (with a finite set of utilities) can be expressed by CGS's: each Γ can be transformed in a CGS $M(\Gamma)$ such that they correspond (in the sense of Definition 15).

The following terms rewrite the specification of best response profiles, Nash equilibria, and the specification of subgame-perfect Nash equilibria from Section 5.3.4. Note that the new specifications use only ATLP operators.

$$\begin{array}{rcl}
BR^T_a(\sigma) & \equiv & (\textbf{set-pl}\ \sigma[\mathbb{A}\mathrm{gt}\setminus\{a\}])\mathbf{Pl}\bigwedge\limits_{v\in U}\Big((\langle\langle a\rangle\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}}) \rightarrow (\textbf{set-pl}\ \sigma)\langle\langle\emptyset\rangle\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}}\Big) \\
NE^T(\sigma) & \equiv & \bigwedge\limits_{a\in\mathbb{A}\mathrm{gt}} BR^T_a(\sigma) \\
SPN^T(\sigma) & \equiv & \langle\langle\emptyset\rangle\rangle\Box NE^T(\sigma)
\end{array}$$

[11] Note the similarity of the scheme below to the characterization of qualitative Nash equilibrium in Section 5.5.3.

Recalling briefly the ideas behind the above specifications, $BR_a^T(\sigma)$ holds iff $\sigma[a]$ is the best response to $\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]$. That is, after we fix the $\mathbb{A}\mathrm{gt} \setminus \{a\}$'s collective strategy to $\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]$, agent a cannot obtain a better temporal pattern of payoffs than by playing $\sigma[a]$. Then, σ is a *Nash equilibrium* if each individual strategy $s[a]$ is the best response to the opponent's strategies $\sigma[\mathbb{A}\mathrm{gt} \setminus \{a\}]$ (cf. [111]). The formalization of a subgame perfect Nash equilibrium is straightforward: We require profile σ to be a Nash equilibrium in all reachable states (seen as initial positions of particular subgames).

The following propositions are simple adaptations of the results from Section 5.3.4.

Proposition 54 *Let Γ be an extensive game with a finite set of utilities. Then the following holds:*

1. $s \in [\![\sigma.NE^{\diamond}(\sigma)]\!]^{\emptyset}_{M(\Gamma)}$ *iff s is a Nash equilibrium in Γ.*
2. $s \in [\![\sigma.SPN^{\diamond}(\sigma)]\!]^{\emptyset}_{M(\Gamma)}$ *iff s is a subgame perfect Nash equilibrium in Γ.*

In Section 5.3.4 we defined a quantitative version of *Pareto optimality* formulated in ATLI. However, as we pointed out, the ATLI formula had exponential length and some counterintuitive implications. Quantification allows to propose a more compact and intuitive specification:

$$PO^T(\sigma) \equiv \forall\sigma' \mathbf{Pl}\Big(\bigwedge_{a\in\mathbb{A}\mathrm{gt}} \bigwedge_{v\in U} \big((\textbf{set-pl}\ \sigma')\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}} \rightarrow (\textbf{set-pl}\ \sigma)\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}}\big) \vee \bigvee_{a\in\mathbb{A}\mathrm{gt}} \bigvee_{v\in U} \big((\textbf{set-pl}\ \sigma)\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}} \wedge \neg(\textbf{set-pl}\ \sigma')\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}}\big)\Big).$$

This definition of Pareto optimality is more intuitive than the one given in Section 5.3.4 because it does not focus on temporal evolution of whole payoff profiles, but rather on the interaction between temporal patterns of individual patterns.

Proposition 55 *Let Γ be an extensive game with a finite set of utilities. Then:*

$$s \in [\![\sigma.PO^{\diamond}(\sigma)]\!]^{\emptyset}_{M(\Gamma)} \text{ iff } s \text{ is Pareto optimal in } \Gamma.$$

Let $\langle x^A, y^{\mathbb{A}\mathrm{gt}\setminus A}\rangle$ be a shorthand for the term $\langle z_1, \ldots, z_k\rangle$ with $z_a = x$ for $a \in A$ and $z_a = y$ otherwise. The following specification, formulated as an $\mathcal{L}^1_{ATLP}$ formula, characterizes the set of strategy profiles that include undominated strategies for agent a:

$$UNDOM^T(\sigma) \equiv \forall\sigma_1\forall\sigma_2\exists\sigma_3$$

$$\mathbf{Pl}\Big(\bigwedge_{v\in U} \big((\textbf{set-pl}\ \langle\sigma_1^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}} \rightarrow (\textbf{set-pl}\ \langle\sigma^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}}\big)$$

$$\vee \bigvee_{v\in U} \big((\textbf{set-pl}\ \langle\sigma^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}} \wedge \neg(\textbf{set-pl}\ \langle\sigma_1^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\setminus\{a\}}\rangle)\langle\!\langle\emptyset\rangle\!\rangle T\mathsf{p}^{\mathsf{v}}_{\mathsf{a}}\big)\Big).$$

Proposition 56 *Let Γ be an extensive game with a finite set of utilities. Then*

$$s \in [\![\sigma.UNDOM^{\diamond}(\sigma)]\!]^{\emptyset}_{M(\Gamma)} \text{ iff } s|_a \text{ is undominated in } \Gamma.$$

5.5.3 General Solution Concepts in $\mathcal{L}^1_{ATLP}$

In this section, we return to the idea of *general solution concepts* from Section 5.3.5 and show how qualitative versions of NE, SPN, PO and UNDOM can be captured in ATLP. Like for temporalized solution concepts, it turns out that their qualitative counterparts can be already specified in $\mathcal{L}^1_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \emptyset)$. That is, we need only one level of nested plausibility updates (and no "hardwired" plausibility terms) to effectively capture classical notions of rationality and extend them to more general games that we study in this paper.

We only consider one "winning condition" per agent to represent agents' preferences, but this view can be naturally extended to full preference lists, as in Section 3.5. In what follows, let $\overrightarrow{\eta} = \langle \eta_1, \dots, \eta_k \rangle$ be a vector of $\mathcal{L}_{ATL}$ path formulae.

Definition 37 (Transforming a CGSP into a NF Game) *Let* $M \in CGSP(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$ *and* $q \in St_M$. *The* associated NF game $\mathcal{S}(M, \overrightarrow{\eta}, q)$ *with respect to* $\overrightarrow{\eta}$ *is given as in Definition 18 with* M *interpreted as a pure CGS by removing* Υ *and* $[\![\cdot]\!]$ *from it.*

Our aim is to define analogues of classical solution concepts (Nash equilibria and such) that are based on explicit "winning conditions" η_i instead of numerical payoffs. We can build on our results from the previous section; we only need to replace temporal patterns of payoffs with the formulae η_i:

$$
\begin{aligned}
BR_a^{\overrightarrow{\eta}}(\sigma) &\equiv (\textbf{set-pl}\ \sigma[\mathbb{A}\mathrm{gt}\backslash\{a\}])\mathbf{Pl}\left(\langle\!\langle a\rangle\!\rangle\eta_a \to (\textbf{set-pl}\ \sigma)\langle\!\langle\emptyset\rangle\!\rangle\eta_a\right)\\
NE^{\overrightarrow{\eta}}(\sigma) &\equiv \bigwedge_{a\in\mathbb{A}\mathrm{gt}} BR_a^{\overrightarrow{\eta}}(\sigma)\\
SPN^{\overrightarrow{\eta}}(\sigma) &\equiv \langle\!\langle\emptyset\rangle\!\rangle\Box NE^{\overrightarrow{\eta}}(\sigma)
\end{aligned}
$$

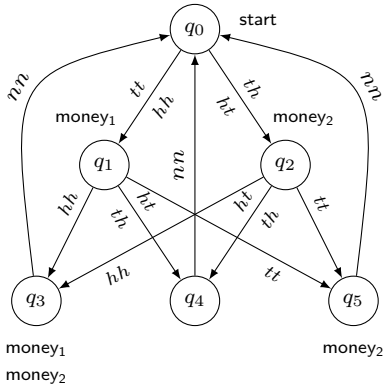
$$
\begin{aligned}
PO^{\overrightarrow{\eta}}(\sigma) \equiv\ & \forall\sigma'\,\mathbf{Pl}\Big(\bigwedge_{a\in\mathbb{A}\mathrm{gt}}((\textbf{set-pl}\ \sigma')\langle\!\langle\emptyset\rangle\!\rangle\eta_a \to (\textbf{set-pl}\ \sigma)\langle\!\langle\emptyset\rangle\!\rangle\eta_a) \vee\\
& \bigvee_{a\in\mathbb{A}\mathrm{gt}}((\textbf{set-pl}\ \sigma)\langle\!\langle\emptyset\rangle\!\rangle\eta_a \wedge \neg(\textbf{set-pl}\ \sigma')\langle\!\langle\emptyset\rangle\!\rangle\eta_a\Big).
\end{aligned}
$$

$$
\begin{aligned}
UNDOM^{\overrightarrow{\eta}}(\sigma) \equiv\ & \forall\sigma_1\forall\sigma_2\exists\sigma_3\,\mathbf{Pl}\\
&\Big(\big((\textbf{set-pl}\ \langle\sigma_1^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\backslash\{a\}}\rangle)\langle\!\langle\emptyset\rangle\!\rangle\eta_a \to (\textbf{set-pl}\ \langle\sigma^{\{a\}}, \sigma_2^{\mathbb{A}\mathrm{gt}\backslash\{a\}}\rangle)\langle\!\langle\emptyset\rangle\!\rangle\eta_a\big)\\
&\vee\big((\textbf{set-pl}\ \langle\sigma^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\backslash\{a\}}\rangle)\langle\!\langle\emptyset\rangle\!\rangle\eta_a \wedge \neg(\textbf{set-pl}\ \langle\sigma_1^{\{a\}}, \sigma_3^{\mathbb{A}\mathrm{gt}\backslash\{a\}}\rangle)\langle\!\langle\emptyset\rangle\!\rangle\eta_a\big)\Big).
\end{aligned}
$$

The intuitions behind these concepts are the same as in the quantitative case. Note that we did not have to include the big conjunctions/disjunctions over all possible utility values in the case of Pareto optimal and undominated strategies. This is because the corresponding NF game can be seen as a game with only *two* possible outcomes per agent.

The following proposition shows that $NE^{\overrightarrow{\eta}}$, $PO^{\overrightarrow{\eta}}$, and $UNDOM^{\overrightarrow{\eta}}$ indeed extend the classical notions of Nash equilibrium, Pareto optimal strategy profile, and undominated strategy.

(A)



(B)

$\eta_1 \backslash \eta_2$	s_{hh}	s_{ht}	s_{th}	s_{tt}
s_{hh}	$\mathbf{1,1}$	$0,0$	$0,1$	$0,1$
s_{ht}	$0,0$	$0,1$	$0,1$	$0,1$
s_{th}	$0,1$	$0,1$	$\mathbf{1,1}$	$0,0$
s_{tt}	$0,1$	$0,1$	$0,0$	$0,1$

Figure 5.7: "Extended matching pennies": (A) CGS M_3; again, action profile xy refers to action x played by player 1 and action y played by 2. (B) Strategies and their outcomes for $\eta_1 \equiv \Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1)$, $\eta_2 \equiv \Diamond\mathsf{money}_2$. Pareto optimal profiles are indicated with bold font, Nash equilibria with grey background.

Proposition 57

1. *The set of Nash equilibrium strategies in $\mathcal{S}(M, \overrightarrow{\eta}, q)$ is given by $[\![\sigma.NE^{\overrightarrow{\eta}}(\sigma)]\!]^q_M$.*
2. *The set of Pareto optimal strategies in $\mathcal{S}(M, \overrightarrow{\eta}, q)$ is given by $[\![\sigma.PO^{\overrightarrow{\eta}}(\sigma)]\!]^q_M$.*
3. *The set of a's undominated strategies in $\mathcal{S}(M, \overrightarrow{\eta}, q)$ is given by $\left([\![\sigma.UNDOM^{\overrightarrow{\eta}}(\sigma)]\!]^q_M\right)|_a$.*

Subgame perfect Nash equilibria cannot be directly related to normal form games, but we can state the following.

Proposition 58 *Let St' be the set of states reachable from q in M. Then, $[\![\sigma.SPN^{\eta}(\sigma)]\!]^q_M = \bigcap_{q \in St'} [\![\sigma.NE^{\eta}(\sigma)]\!]^q_M$.*

Example 32 (Extended matching pennies) *In Figure 5.7 we consider a slightly more complex version of the* asymmetric matching pennies game *presented in Figure 5.5. The new game consists of two phases (played* ad infinitum*). Firstly, player 1 wins some money if the sides of the pennies match, otherwise the money goes to player 2. In the second phase, both win a prize if both show heads; if they both show tails, only player 2 wins. If they show different sides, nobody wins.*

We denote particular strategies as $s_{\alpha_1\alpha_2}$, where α_1 is the action played at state q_0, and α_2 is the action played at states q_1, q_2 (it is not necessary to consider strategies that specify different actions in q_1 and q_2, since the outgoing transitions in q_2 are exact copies of those in q_1). Note that every combination of strategies (i.e., every strategy profile) determines a single temporal path. For example, if agent 1 plays s_{ht} and agent 2 plays s_{tt}, then they both ensure the (infinite) temporal path $q_0q_2q_5q_0q_2q_5\ldots$

Let us additionally assume that the winning conditions are: $\eta_1 \equiv \Box(\neg\mathsf{start} \rightarrow \mathsf{money}_1)$ for player 1 and $\eta_2 \equiv \Diamond\mathsf{money}_2$ for player 2. That is, agent 1 is only

happy if she gets money all the time (whenever possible). Agent 2 is more minimalistic: it is sufficient for him to win money once, sometime in the future. So, for instance, the play that results from strategy profile $\langle s_{ht}, s_{tt}\rangle$ satisfies the second player, but not the first one. This way, it is easy to construct a table of binary payoffs that indicates which strategy profiles are "winning" for whom, like the table in Figure 5.7B. Now, we can for instance observe that profile $\langle s_{ht}, s_{tt}\rangle$ is a Nash equilibrium (player 1 cannot make herself happy by unilaterally changing her strategy), but it is not Pareto optimal ($\langle s_{hh}, s_{hh}\rangle$ and $\langle s_{th}, s_{th}\rangle$ yield strictly better payoff profiles). As before, the CGS M_3 in Figure 5.7A can be seen as a CGSP by adding $\Upsilon = \Sigma$ and $\Omega = \emptyset$. Now, we have that:

- $[\![\sigma.NE^{\eta_1,\eta_2}(\sigma)]\!]^{q_0}_{M_3} = \{\langle s_{hh}, s_{hh}\rangle, \langle s_{hh}, s_{tt}\rangle, \langle s_{ht}, s_{ht}\rangle, \langle s_{ht}, s_{tt}\rangle, \langle s_{th}, s_{ht}\rangle, \langle s_{th}, s_{th}\rangle, \langle s_{tt}, s_{ht}\rangle, \langle s_{tt}, s_{tt}\rangle\}$, *and*
- $[\![\sigma.PO^{\eta_1,\eta_2}(\sigma)]\!]^{q_0}_{M_3} = \{\langle s_{hh}, s_{hh}\rangle, \langle s_{th}, s_{th}\rangle\}$.

Suppose that agent 1 *wants money always, and* 2 *wants money eventually, and only Pareto optimal Nash equilibria are played. Then, agent* 1 *is bound to get money at the beginning of each round of the game. Formally:*

$$M_3, q_0 \models (\textbf{\textit{set-pl}}\ \sigma.NE^{\eta_1,\eta_2}(\sigma))(\textbf{\textit{refn-pl}}\ \sigma.PO^{\eta_1,\eta_2}(\sigma))\mathbf{Pl}\,(\mathsf{start} \rightarrow \langle\!\langle\emptyset\rangle\!\rangle \bigcirc \mathsf{money}_1).$$

In ATLP, we can also describe relationships between different solution concepts in a CGS. For example, in the "extended matching pennies" game, all Pareto optimal profiles happen to be in Nash equilibrium, which is equivalent to the following formula:

$$(\textbf{set-pl}\ \sigma.PO^{\eta_1,\eta_2}(\sigma))(\textbf{refn-pl}\ \sigma.\neg NE^{\eta_1,\eta_2}(\sigma))\mathbf{Pl}\,\neg\langle\!\langle\mathbb{A}\mathrm{gt}\rangle\!\rangle \bigcirc \top,$$

and the formula does indeed hold in M_3, q_0.

5.6 Model Checking ATLP

In this section we discuss the model checking complexity of ATLP. The *model checking problem* refers to the question whether a given formula holds in a given model and state. The size of the input is usually measured in the number of transitions in the model (m) and the length of the formula (l). Note that the problem of checking ATLP with respect to the size of the *whole* CGSP (including the plausibility set Υ), is trivially linear in the size of the model: The model size is *exponential with respect to the number of states and transitions*. Hence, model checking CGSP's does not make sense if the set of plausible strategies is stored explicitly. The set should be stored implicitly; for instance, by means of some decision procedure. We will assume throughout this section that the plausibility set Υ does not discriminate any strategy profiles (i.e., all strategy profiles are initially plausible), and actual plausibility assumptions must be specified in the object language through (simple or complex) plausibility terms.

The same remark applies to the denotations of primitive ("hard-wired") plausibility terms. In this respect, we will consider two subclasses of CGSP's in which the representation of plausibility assumptions of plausibility assumptions does not overwhelm the complexity of the rest of the input –

namely, pure concurrent game structures and so called "well-behaved" CGSP's. In pure CGS's, plausibility terms and their denotations are simply absent. In well-behaved CGS's, we put a limit on the complexity of the *plausibility check*, i.e., the computational resources needed to determine whether a given strategy is plausible according to a given plausibility term and plausibility mapping.

Definition 38 (CGS as CGSP) *As before, we will take each CGS to be an implicit representation of CGSP where all strategy profiles are initially plausible ($\Upsilon = \Sigma$) and there are no "hardwired" plausibility terms ($\Omega = \emptyset$).*

Definition 39 (Well-Behaved CGSP) *A CGSP M is called* well-behaved *if, and only if,*

1. $\Upsilon_M = \Sigma$*: all the strategy profiles are plausible in M;*
2. *There is an* **NP***-algorithm (with respect to l and m) which determines whether $s \in [\![\omega]\!]^q_M$ for every state $q \in St_M$, strategy profile $s \in \Sigma$, and plausibility term $\omega \in \Omega$.*

Remark 21 *We note that, if a list (or several alternative lists) of plausible strategy profiles is given* explicitly in the model *(via the plausibility set Υ and/or the denotations of abstract plausibility terms ω from Section 5.4), then the problem of guessing an appropriate strategy from such a list is in* **NP** *(memoryless strategies have polynomial size with respect to m). Consequently, we assume that, if such a list is given explicitly, that it is stored* outside *the model.*

We begin our study with the complexity of model checking the basic language $\mathcal{L}^{\text{base}}_{ATLP}$ in Section 5.6.1. Then, we investigate the complexity for the intermediate language $\mathcal{L}_{ATLP^{ATLI}}$ (Section 5.6.2). It turns out that the problem is in both cases $\mathbf{\Delta^P_3}$-complete in general, which seems in line with existing results on the complexity of solving games. In particular, it is known that if both players in a 2-player imperfect information game have imperfect recall, and chance moves are allowed, then the problem of finding a maxmin pure strategy is $\mathbf{\Sigma^P_2}$-complete [89].[12] That is, there are established results within game theory which show that reasoning about the outcome of a game where the strategies of both parties are restricted cannot be easier than $\mathbf{\Sigma^P_2}$ (resp. $\mathbf{\Delta^P_3}$ when nesting of game specifications is allowed). In the light of this, our complexity results are not as pessimistic as they seem, especially as ATLP allows specification of much more diverse restrictions than those imposed by imperfect information in 2-player turn-based games.[13]

Moreover, we show in Sections 5.6.1 and 5.6.2 that model checking $\mathcal{L}^{\text{base}}_{ATLP}$ and $\mathcal{L}_{ATLP^{ATLI}}$ is $\mathbf{\Delta^P_2}$-complete if only the proponents' strategies are restricted. This, again, corresponds to some well-known **NP**-hardness results for solving extensive games with imperfect information and recall [28, 48, 89].

Finally, in Section 5.6.3 we study the model checking complexity of $\mathcal{L}^k_{ATLP}$ and $\mathcal{L}^\infty_{ATLP}$. We summarize the results in Section 5.6.4.

[12] Note that strategic operators can be nested in an ATLP formula, thus specifying a sequence of games, with the outcome of each game depending on the previous ones—and solving such games requires adaptive calls to a $\mathbf{\Sigma^P_2}$ oracle.

[13] In particular, imperfect information strategies (sometimes called *uniform* strategies) can be characterized in ATLP for a relevant subclass of models, cf. Section 5.6.1.

function $mcheckATLP(M, q, \varphi)$**;**
Model checking ATLP: the main function.

- Return $mcheck(M, q, \varphi, \emptyset, \emptyset)$;

function $mcheck(M, q, \varphi, \overrightarrow{\omega}, B)$**;**
Returns "true" iff φ plausibly holds in M, q. The current plausibility assumptions are specified by a sequence $\overrightarrow{\omega} = [\langle\omega_1, q_1\rangle, \ldots, \langle\omega_n, q_n\rangle]$ of plausibility terms with interpretation points. The set of agents which are assumed to play rational are denoted by B.

cases $\varphi \equiv p$**,** $\varphi \equiv \neg\psi$**,** $\varphi \equiv \psi_1 \wedge \psi_2$ **:** proceed as usual;
case $\varphi \equiv$ (**set-pl** ω')ψ **:** return($mcheck(M, q, \psi, [\langle\omega', q\rangle], B)$);
case $\varphi \equiv$ (**refn-pl** ω')ψ **:** return($mcheck(M, q, \psi, \overrightarrow{\omega} \oplus \langle\omega', q\rangle, B)$);
case $\varphi \equiv$ **Pl** $_A\psi$ **:** return($mcheck(M, q, \psi, \overrightarrow{\omega}, A)$);
case $\varphi \equiv \langle\langle A\rangle\rangle\bigcirc\psi$**, where** ψ **includes some** $\langle\langle B\rangle\rangle$ **:** Label all $q' \in St$, in which $mcheck(M, q, \psi, \overrightarrow{\omega}, B)$ returns "true", with a new proposition yes. Return $mcheck(M, q, \langle\langle A\rangle\rangle\bigcirc$ yes$, \overrightarrow{\omega}, B)$;
case $\varphi \equiv \langle\langle A\rangle\rangle\bigcirc\psi$**, where** ψ **includes no** $\langle\langle C\rangle\rangle$ **:** Remove all operators **Pl** , **Ph** , (**set-pl** $\cdot$) from ψ (they are irrelevant, as no cooperation modality comes further), yielding ψ'. Return $solve(M, q, \langle\langle A\rangle\rangle\bigcirc\psi', \overrightarrow{\omega}, B)$;
cases $\langle\langle A\rangle\rangle\Box\psi$ **and** $\langle\langle A\rangle\rangle\psi_1\,\mathcal{U}\,\psi_2$ **:** analogously ;
end case

function $solve(M, q, \varphi, \overrightarrow{\omega}, B)$**;**
Returns "true" iff φ holds in M, q under plausibility assumptions specified by $\overrightarrow{\omega}$ and applied to B. We assume that $\varphi \equiv \langle\langle A\rangle\rangle\Box\psi$, where ψ is a propositional formula, i.e., it includes no $\langle\langle B\rangle\rangle$, **Pl** , **Ph** , (**set-pl** $\cdot$).

- Label all $q' \in St$, in which ψ holds, with a new proposition yes;
- Guess a strategy profile s;
- **if** $plausiblestrat(s, M, \overrightarrow{\omega}, B)$
  then return(not $beatable(s[A], M, q, \langle\langle A\rangle\rangle\Box$yes$, \overrightarrow{\omega}, B)$);
  else return(false);

Figure 5.8: Model checking ATLP

5.6.1 Model Checking $\mathcal{L}^{\mathbf{base}}_{ATLP}$

In this section we show that model checking $\mathcal{L}^{\text{base}}_{ATLP}$ is $\mathbf{\Delta_3^P}$-complete in general, and $\mathbf{\Delta_2^P}$-complete when only the proponents' strategies are restricted. Moreover, model checking $\mathcal{L}^{\text{base}}_{ATLP}$ over *rectangular models* and models with *bounded plausibility sets* can be done in polynomial time.

Model Checking $\mathcal{L}^{\mathbf{base}}_{ATLP}$: Upper Bounds

Well-behaved CGSP. A detailed algorithm for model checking $\mathcal{L}^{\text{base}}_{ATLP}$ formulae in well- behaved concurrent game structures with plausibility is presented in Figure 5.8. Apart from model M, state q, and formula φ to be checked, the input includes a plausibility specification vector $\overrightarrow{\omega}$ and a set

function $beatable(s_A, M, q, \langle\langle A\rangle\rangle\gamma, \overrightarrow{\omega}, B)$**;**
Returns "true" iff the opponents can beat s_A so that it does not enforce γ in M, q under plausibility assumptions specified by $\overrightarrow{\omega}$ and imposed on B. The path formula γ is of the form $\bigcirc\psi$, $\Box\psi$, $\psi\,\mathcal{U}\,\psi'$ with propositional ψ, ψ'.

- Guess a strategy profile t;
- **if** $plausiblestrat(t, M, \overrightarrow{\omega}, B)$ and $t|_A = s_A$ **then**
  - $M' :=$ "trim" M, removing all transitions that cannot occur when $t|_B$ is executed;
  - return($mcheck_{CTL}(M', q, \neg\mathsf{A}\gamma)$);

  else return(false);

function $plausiblestrat(s, M, \overrightarrow{\omega}, B)$**;**
Checks whether B's part of strategy profile s is part of some profile in $\bigcap_{\langle\omega,q\rangle\in\overrightarrow{\omega}} [\![\omega]\!]^q_M$.

- return true if $s|_B \in \bigcap_{\langle\omega,q\rangle\in\overrightarrow{\omega}} [\![\omega]\!]^q_M|_B$; and false otherwise.

Figure 5.9: Model checking ATLP, ctd.

B of agents which are assumed to play rationally. The plausibility vector $\overrightarrow{\omega} = [\langle\omega_1, q_1\rangle, \ldots, \langle\omega_n, q_n\rangle]$ is a sequence of plausibility terms together with states at which the terms are evaluated; this is because we need to keep track of applications of the refinement operators (**refn-pl** $\cdot$). The intuition is that the vector represents the incremental plausibility updates. Moreover, by $[\langle\omega_1, q_1\rangle, \ldots, \langle\omega_n, q_n\rangle]\oplus\langle\omega, q\rangle$ we denote the vector $[\langle\omega_1, q_1\rangle, \ldots, \langle\omega_n, q_n\rangle, \langle\omega, q\rangle]$.

CTL model checking is linear in the number of transitions in the model and the length of the formula [30], so as long as $plausiblestrat(s, M, q, \omega, B)$ can be computed in polynomial time, we get that $mcheckATLP$ runs in time $\mathbf{\Delta_3^P}$, i.e., the algorithm can be implemented as a deterministic Turing Machine making adaptive calls to an oracle of range $\mathbf{\Sigma_2^P} = \mathbf{NP^{NP}}$. In fact, it suffices to require that $plausiblestrat(s, M, q, \omega, B)$ can be computed in *nondeterministic* polynomial time, as the witness for $plausiblestrat$ can be guessed together with the strategy profile s in function $solve$, and with the strategy profile t in function $beatable$, respectively. The intersection of plausibility terms can also be neglected as the vector of plausibility terms can contain at most l terms (length of the formula). Schematically, we can describe the main part of the algorithm by $\exists s\neg(\exists t)$: s is guessed first, then t is guessed (and its answer is negated, so we have $\exists s\forall t$). This schematic view will be useful in Section 5.6.3 to give an intuition about the complexity of nested formulae together with quantification over strategic terms.

Proposition 59 *Let M be a well-behaved CGSP, q a state in M, and φ a formula of $\mathcal{L}^{base}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$. Then $M, q \models \varphi$ iff $mcheckATLP(M, q, \varphi)$. The algorithm runs in time $\mathbf{\Delta_3^P}$ with respect to the number of transitions in the model and the length of the formula.*

Proof in Appendix 5.11.1.

Note that the requirement that the set of plausible strategies is given by Σ is not a real restriction. Specific plausibility specification can always be set using operator (**set-pl** $\cdot$), by adding a new plausibility term that denotes the desired set of strategy profiles. The only restriction is that inclusion in the set must be verifiable in nondeterministic polynomial time.

Finally, we observe that the complexity can be improved if only the strategies of the proponents are restricted.

Proposition 60 *Let γ be an $\mathcal{L}^{base}_{ATLP}$ path formula without cooperation modalities. Then the model checking problem for formulae of the form $\mathbf{Pl}_A\langle\langle A\rangle\rangle\gamma$ is in $\mathbf{\Delta_2^P}$ (instead of $\mathbf{\Delta_3^P}$).*

Proof sketch We consider the case $\varphi \equiv \langle\langle A\rangle\rangle\bigcirc\psi$, where ψ includes no $\langle\langle C\rangle\rangle$. In $solve$ a plausible strategy s_A for A is guessed (**NP**-call). Then, in function $beatable$ the model is directly trimmed according to s_A (without guessing another profile t) and the CTL model checking algorithm is executed. In this case, function $beatable$ can be executed in polynomial time. ■

Corollary 6 *Let $\varphi \in \mathcal{L}^{base}_{ATLP}$. If for each cooperation modality $\langle\langle A\rangle\rangle$ occurring in φ it is specified that only agents A' where $A' \subseteq A$ play plausibly then model checking is in $\mathbf{\Delta_2^P}$.*

Pure CGS. This is a somewhat degenerate case because in $\mathcal{L}^{\text{base}}_{ATLP}$ only primitive plausibility terms can be used. With no such terms, (**set-pl** $\cdot$) and (**refn-pl** $\cdot$) operators cannot be used, so all strategy profiles will be considered plausible in the evaluation of every subformula. In consequence, model $\mathcal{L}^{\text{base}}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \emptyset)$ can be done in the same way as for ATL. Since model checking ATL lies in **P** [8] we get the following result.

Proposition 61 *Let M be a CGS, q a state in M, and $\varphi \in \mathcal{L}^{base}_{ATLP}(\mathbb{A}\text{gt}, \Pi, \emptyset)$. Model checking φ in M, q is in* **P** *with respect to the number of transitions in the model and the length of the formula.*

Proof Remove all $\mathbf{Pl}_A$ operators from φ and check whether $M'q, \models_{ATL} \varphi$ where M' is the CGS obtained from M by leaving out Υ, Ω, and $[\![\cdot]\!]$. ■

Special Classes of Models. We will now consider the special case in which each plausibility term refers to at most polynomially many strategies.

Definition 40 (Bounded Models $\mathfrak{M}^c$) *Given a fixed constant $c \in \mathbb{N}$ we consider the class $\mathfrak{M}^c \subseteq CGSP(\mathbb{A}\text{gt}, \Pi, \Omega)$ of models such that for all $M \in \mathfrak{M}^c$, $\omega \in \Omega_M$, and $q \in St_M$ it holds that $|[\![\omega]\!]^q_M| \leq l^c \cdot m^c$ where l (resp. m) denotes the length of the input formula (resp. number of transitions of M).*

Proposition 62 *Let $c \in \mathbb{N}$ be a constant. Model checking $\mathcal{L}^{base}_{ATLP}$ formulae with respect to the class of well-behaved bounded models $\mathfrak{M}^c$ can be done in polynomial time with respect to the number of transitions in the model and the length of the formula.*

Proof in Appendix 5.11.1.

Even with arbitrarily many strategies the complexity can be improved if the set of plausible profiles has a specific structure, namely if the set can be (and is) represented in a *rectangular* way. Intuitively, such a set of profiles can be represented by behavioral constraints [132]. That is, we restrict the actions that can be performed independently for each state and agent, and then consider all strategy profiles generated from the constrained repertoire of actions.

Definition 41 (Rectangularity, $\mathfrak{M}^{\mathbf{rect}}$) *Let $S_a \subseteq \Sigma_a$ be a set of strategies of agent a. We say that S_a is* rectangular *if it is represented by a function $d'_a : St_M \to 2^{Act}$ such that for all states $q \in St_M$ it holds that $d'_a(q) \subseteq d_a(q)$; then, S_a is taken to be the set $\{s_a \in \Sigma_a \mid \forall q \in St_M\ (s_a(q) \in d'_a(q))\}$.*

A set of collective strategies (resp. strategy profiles) $S_A \subseteq \Sigma_A$ is rectangular if it represented as a collection of rectangular sets of individual strategies. Then, S_A is to the Cartesian product of the individual sets, i.e., $S_A = \prod_{a \in A} S_a$.

A set of plausibility terms Ω is rectangular in a model M if all terms in $\omega \in \Omega$ have rectangular denotations $[\![\omega]\!]^q_M$. Finally, we say that a CGSP M is rectangular if the set Υ_M is rectangular and terms Ω are rectangular in M. We denote the class of such models by $\mathfrak{M}^{rect}$.

Note, for example, that each Σ_A is rectangular.

Proposition 63 *Model checking $\mathcal{L}^{base}_{ATLP}$ formulae in the class $\mathfrak{M}^{rect}$ can be done in* **P** *with respect to the number of transitions in the model and the length of the formula.*

Proof The algorithm is very simple; we present the procedure for $\varphi \equiv \langle\langle A \rangle\rangle \Box \psi$ being in the scope of (**set-pl** ω) and $\mathbf{Pl}_B$. Other cases are analogous.

Firstly, we model-check (**set-pl** ω)$\mathbf{Pl}_B\psi$ recursively and label the states where the answer was "true" with a new proposition yes. Then, we take $[\![\omega]\!]^q_M$ (recall that it is represented in a rectangular way, i.e., by function $d' : \mathbb{A}\mathrm{gt} \times St \to 2^{Act}$), and replace function d in M by d'' such that $d''(a,q) = d'(a,q)$ for $a \in B$ and $d''(a,q) = d(a,q)$ for $a \notin B$. Finally, we use any ATL model checker to model-check $\langle\langle A \rangle\rangle \Box$yes in the resulting model, and return the answer. ■

We observe that strategic combinations of rectangular plausibility terms are also rectangular. In consequence, the results extends to $\mathcal{L}^0_{ATLP}$ in a straightforward way, which will prove useful in Section 5.6.3.[14]

Lemma 11 *If $S \subseteq \Sigma_a$ (resp. $S \subseteq \Sigma_A$) contains only a single strategy (resp. strategy profile) then it is rectangular.*

Lemma 12 *Let Ω be a rectangular set of plausibility terms, then $\tau(\Omega)$ is rectangular as well.*

Corollary 7 *Model checking $\mathcal{L}^0_{ATLP}$ formulae in the class $\mathfrak{M}^{rect}$ can be done in* **P** *with respect to the number of transitions in the model and the length of the formula.*

[14] Recall, that $\mathcal{L}^0_{ATLP}$ consists of all base formulae in which plausibility terms form $\tau(\Omega)$ can be used (instead of plain terms from Ω only).

Model Checking $\mathcal{L}^{\text{base}}_{ATLP}$: Hardness and Completeness

Well-behaved CGSP. We prove $\mathbf{\Delta^P_3}$-hardness through a reduction of SNSAT$_2$, a typical $\mathbf{\Delta^P_3}$-complete variant of the Boolean satisfiability problem. The reduction is done in two steps.

1. Firstly, we define a modification of ATL$_{ir}$ [121], in which *all* agents are required to play only uniform strategies. We call it "uniform ATL$_{ir}$" (ATL$^u_{ir}$ in short), and show that model checking ATL$^u_{ir}$ is $\mathbf{\Delta^P_3}$-complete by means of a polynomial reduction of SNSAT$_2$ to ATL$^u_{ir}$ model checking.

2. Then, we point out that each formula and model of ATL$^u_{ir}$ can be equivalently translated (in polynomial time) to a CGSP and a formula of $\mathcal{L}^{\text{base}}_{ATLP}$, thus yielding a polynomial reduction of SNSAT$_2$ to model checking $\mathcal{L}^{\text{base}}_{ATLP}$.

Parts of our construction reuse techniques presented in [52, 81, 70, 82].

In "uniform ATL$_{ir}$" (ATL$^u_{ir}$), where we assume that all the players have limited information about the current state, and each agent can only use *uniform* strategies (i.e., ones that assign same choices in indistinguishable states). The syntax of ATL$^u_{ir}$ is the same as that of ATL, only cooperation modalities are annotated with additional tags ir and u to indicate the imperfect **i**nformation and **r**ecall, and **u**niformity of all agents' strategies. The semantics of ATL$^u_{ir}$ is defined over *concurrent epistemic game structures* (CEGS), i.e. CGS extended with epistemic relations that represent indistinguishability of states for agents. Details of the semantics and more thorough presentation can be found in Appendix 5.9. The following proposition summarizes the complexity results from Appendix 5.9.2.

Proposition 64 *Model checking ATL$^u_{ir}$ is $\mathbf{\Delta^P_3}$-complete with respect to the number of transitions in the model and the length of the formula.*

Remark 22 *We have thus proven that checking strategic abilities when* all *players are required to play uniformly is $\mathbf{\Delta^P_3}$-complete (that is, harder than ability compared with the worst line of events captured by ATL$_{ir}$ formulae, which is "only" $\mathbf{\Delta^P_2}$-complete). We believe it is an interesting result with respect to verification of various kinds of agents' abilities under incomplete information. We note that the result from [89] for extensive games with incomplete information can be seen as a specific case of our result, at least in the class of games with binary payoffs.*

Now we show how ATL$^u_{ir}$ model checking can be reduced to model checking of $\mathcal{L}^{\text{base}}_{ATLP}$. We are given a CEGS M, a state q in M, and an ATL$^u_{ir}$ formula φ. Let Σ^u be the set of all uniform strategy profiles in M. We take CGSP M' as M (sans epistemic relations) extended with plausibility mapping $[\![\cdot]\!]$ such that $[\![\omega]\!]^q = \Sigma^u$. Then:

$$M, q \models_{\text{ATL}^u_{ir}} \langle\!\langle A \rangle\!\rangle^u_{ir} \varphi \quad \text{iff} \quad M', q \models_{\text{ATLP}} (\textbf{set-pl}\ \omega)\mathbf{Pl}\, \langle\!\langle A \rangle\!\rangle \varphi,$$

which completes the reduction.

Remark 23 *We note in passing that, technically, the size of the resulting model M' is not entirely polynomial. M' includes the plausibility set Υ, which is exponential in the number of states in M (since it is equal to the the set of all uniform strategy profiles in M). This is of course the case when we want to store Υ explicitly. However, checking if a strategy profile is uniform can be done in time linear wrt the number of states in M, so an* implicit *representation of Υ (e.g., the checking procedure itself) requires only linear space.*

As a result of this and Proposition 59, we obtain the following theorem.

Theorem 20 *Model checking $\mathcal{L}^{base}_{ATLP}$ for well-behaved CGSP's is $\mathbf{\Delta^P_3}$-complete with respect to the number of transitions in the model and the length of the formula.*

For the special case when only the proponents have to follow plausible strategies, a reduction from model checking ATL$_{ir}$ (instead of ATL$^u_{ir}$) is sufficient. Since model checking ATL$_{ir}$ is $\mathbf{\Delta^P_2}$-complete [121, 82], we get the following.

Theorem 21 *Let $\mathcal{L}$ the subset of $\mathcal{L}^{base}_{ATLP}$ in which every cooperation modality $\langle\langle A\rangle\rangle$ occurs in the scope of $\mathbf{Pl}_B$ with $B \subseteq A$. Then, model checking $\mathcal{L}$ in the class of well-behaved CGSP's is $\mathbf{\Delta^P_2}$-complete.*

Proof sketch The inclusion in $\mathbf{\Delta^P_2}$ has been already shown in Section 5.6.1. We prove the lower bound by a reduction of model checking Schobbens' ATL$_{ir}$ [121] to model checking of our sublanguage $\mathcal{L}$. Let M be a CEGS, q a state in M, and $\varphi \equiv \langle\langle A\rangle\rangle_{ir}\gamma$ a formula of ATL$_{ir}$. Moreover, let Σ^u_A be the set of all strategy profiles in M that are uniform for A. We take CGSP M' as M (sans epistemic relations) extended with plausibility mapping $[\![\cdot]\!]$ such that $[\![\omega]\!]^q = \Sigma^u_A$. Then:

$$M, q \models_{\text{ATL}_{ir}} \langle\langle A\rangle\rangle_{ir}\gamma \quad \text{iff} \quad M', q \models_{\text{ATLP}} (\textbf{set-pl}\ \omega)\mathbf{Pl}\,\langle\langle A\rangle\rangle\gamma,$$

which completes the reduction. ∎

Pure CGS and Special Classes of Models. In order to show lower bounds for model checking $\mathcal{L}^{\text{base}}_{ATLP}$ for pure concurrent game structures, well-behaved bounded models, and rectangular models, we observe that ATL is a subset of $\mathcal{L}^{\text{base}}_{ATLP}$ even if the latter does not use plausibility terms – and model checking ATL is **P**-complete [8]. Thus, we conclude with the following.

Theorem 22 *Let $c \in \mathbb{N}$ be a constant. Model checking $\mathcal{L}^{base}_{ATLP}$ with respect to well-behaved bounded models $\mathfrak{M}^c$, rectangular models $\mathfrak{M}^{rect}$, and pure CGS's is* **P***-complete.*

5.6.2 Model Checking $\mathcal{L}_{ATLP^{ATLI}}$

Here, we show that model checking ATLP with plausibility terms based on ATLI is also $\mathbf{\Delta^P_3}$-complete. Note that the only primitive terms occurring in formulae of $ATLP^{ATLI}$ are used to simulate strategic terms of ATLI (which denote individual strategies of particular agents. Thus, the results in this section refer to model checking with rectangular CGSP's.

Model Checking $\mathcal{L}_{ATLP^{ATLI}}$: Upper Bound

The algorithm in Figure 5.8 uses abstract plausibility terms but it can also be used for ATLI-based plausibility terms presented in Section 5.4.3. In [85] it was shown that the model checking problem for ATLI is polynomial with respect to the number of transitions and length of the formula. Thus, we get another immediate corollary of Proposition 59.

Proposition 65 *Model checking ATLP with ATLI-based plausibility terms in rectangular well-behaved CGSP's is in* $\mathbf{\Delta_3^P}$ *with respect to the number of transitions in the model and the length of the formula.*

In Section 5.4.4 we have used $\mathcal{L}^1_{ATLP}$ formulae to characterize game theoretic solution concepts. For this purpose it was not necessary to have hardwired plausibility terms in the language. Indeed, the absence of such terms positively influences the model checking complexity of higher levels of ATLP.

Model Checking $\mathcal{L}_{ATLP^{ATLI}}$: Hardness and Completeness

Like in Section 5.6.1, we show the lower bound by a reduction from model checking ATL^u_{ir}. That is, we demonstrate how uniformity of strategy profiles can be characterized by formulae of ATLI for a relevant class of concurrent game structures. The actual reduction is quite technical and can be found in Appendix 5.10. The following result is an immediate corollary of Proposition 67, presented in Appendix 5.10.

Theorem 23 *Model checking* $\mathcal{L}^{base}_{ATLP}$ *with ATLI-based plausibility terms is* $\mathbf{\Delta_3^P}$*-complete with respect to the number of transitions in the model and the length of the formula.*

Moreover, if plausibility restrictions apply only to proponents, then the complexity improves (the proof is analogous to Theorem 21).

Theorem 24 *Let* $\mathcal{L}$ *the subset of* $\mathcal{L}_{ATLP^{ATLI}}$ *in which every cooperation modality* $\langle\langle A\rangle\rangle$ *occurs in the scope of* $\mathbf{Pl}_B$ *with* $B \subseteq A$*. Then, model checking* $\mathcal{L}$ *in the class of well-behaved rectangular CGSP's is* $\mathbf{\Delta_2^P}$*-complete.*

Proof sketch We prove the lower bound (again) by a reduction of model checking ATL_{ir} to model checking $\mathcal{L}$. The reduction is very similar to the one shown in Appendix 5.10 except that only the "verifier" decides upon the values of the propositions (cf. [81]). ■

5.6.3 Model Checking $\mathcal{L}^k_{ATLP}$

In this section we present our results regarding the model checking complexity of the full logic $\mathcal{L}_{ATLP}$. The complexity depends on both the nesting level of ATLP formulae and on the structure and alternations of strategic quantifiers. Before we state our results we introduce some additional definitions needed to classify such complex formulae.

Classifying $\mathcal{L}_{ATLP}$ Formulae: Some Definitions

The complexity of model checking formulae in $\mathcal{L}_{ATLP}$ does not only depend on the actual nesting depth of plausibility terms but also on the structure of strategic quantifiers used inside (**set-pl** $\cdot$) and (**refn-pl** $\cdot$) operators. The latter structure is quite complex and cannot solely be described by the number of quantifiers. Often, a specific position of quantifiers can be used to combine two "guessing" phases, improving complexity.

Firstly, not the number of quantifiers is important but rather the number of alternations. We introduce function $\textsc{Alt} : \{\exists, \forall\}^+ \to \{\exists, \forall\}^+$ which modifies a word over $\{\exists, \forall\}$ such that each quantifier following a quantifier of the same type is removed; for example, $\textsc{Alt}(\exists\forall\forall\forall\exists\forall) = \exists\forall\exists\forall$. Moreover, existential quantifiers at the beginning and end of a quantifier series can, under some conditions, be ignored without changing the model checking complexity. For example, let us assume that the first quantifier is existential. Then it follows a guess of the proponents (resp. opponents) strategy and both guesses can be combined. Analogously, an existential quantifier at the end usually follows another existential guess. To take these issues into account, we define function $\textsc{RAlt} : \{\exists, \forall\}^+ \to \mathbb{Z}$ that counts the number of the *relevant alternations of quantifiers* in a sequence:

$$\textsc{RAlt}(\overrightarrow{Q}) = \begin{cases} n & \text{if } \textsc{Alt}(\overrightarrow{Q}) = Q_1 \dots Q_n \text{ and } Q_1 \neq \exists \neq Q_n; \\ n-1 & \text{if } \textsc{Alt}(\overrightarrow{Q}) = Q_1 \dots Q_n \text{ and } Q_1 = \exists \text{ xor } Q_n = \exists; \\ n-2 & \text{if } \textsc{Alt}(\overrightarrow{Q}) = Q_1 \dots Q_n \text{ and } Q_1 = \exists = Q_n \text{ and } n > 2; \\ -1 & \text{else.} \end{cases}$$

Function RALT characterizes the "hardness" of the outermost level in a given term. The next two functions take into account the recursive structure of terms, due to possibly nested (**set-pl** $\cdot$) or (**refn-pl** $\cdot$) operators. Firstly, $\mathcal{UO}(\varphi)$ returns the set of all the *update operations* (**set-pl** ω) and (**refn-pl** ω) within formula φ. Secondly, *ql* takes a set of update operations and returns the *quantifier level* in these operations as follows:

$$\mathit{ql}(S) = \begin{cases} \max_{s \in S} \mathit{ql}(\{s\}) & \text{if } |S| > 1 \\ \mathit{ql}(\mathcal{UO}(\varphi')) & \text{if } S = \{(\textbf{Op } \sigma.\varphi')\} \\ \textsc{RAlt}(Q_1 \dots Q_n) + \mathit{ql}(\mathcal{UO}(\varphi')) & \text{if } S = \{(\textbf{Op } \sigma.Q_1\sigma_1 \dots Q_n\sigma_n\varphi')\} \\ & \text{and } (\varphi' \notin \mathcal{L}^0_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \mathcal{V}\mathit{ar}, \\ & \mathcal{V}\mathit{ar}) \text{ or } Q_n = \forall) \\ \textsc{RAlt}(Q_1 \dots Q_n) + \mathit{ql}(\mathcal{UO}(\varphi')) + 1 & \text{if } S = \{(\textbf{Op } \sigma.Q_1\sigma_1 \dots Q_n\sigma_n\varphi')\} \\ & \text{and } \varphi' \in \mathcal{L}^0_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \mathcal{V}\mathit{ar}, \\ & \mathcal{V}\mathit{ar}) \text{ and } Q_n = \exists \\ 0 & \text{otherwise} \end{cases}$$

where (**Op** $\cdot$) is either (**set-pl** $\cdot$) or (**refn-pl** $\cdot$).

The intuition behind *ql* is that it determines the maximal sum of relevant alternations in each sequence of nested update operators (**set-pl** $\cdot$) and/or (**refn-pl** $\cdot$). Intuitively, the nested operators represent a tree. Given an $\mathcal{L}^k_{ATLP}$ formula we add arcs from the root of the tree to nodes representing

update operators operators in the kth level. Then, from such a new node representing (**set-pl** ω) or (**refn-pl** ω), we add arcs to nodes representing update operators inside ω (i.e., on the $k-1$th level) and so on. Leaves of the tree consist of nodes representing operators whose terms contain no further update operators. Now, each node represented by e.g. (**set-pl** $\sigma.Q_1\sigma_1 \dots Q_n\sigma_n\varphi'$) is labeled by $\text{RALT}(Q_1 \dots Q_n)$. Function *ql* returns the maximal sum of such numbers along all paths from the root to some leaf.

Given an operator (**set-pl** $\sigma.Q_1\sigma_1 \dots Q_n\sigma_n\varphi'$) on the second to last level without hard-wired plausibility term (i.e., for $\varphi' \in \mathcal{L}^0_{ATLP}(\mathbb{A}\text{gt}, \Pi, \mathcal{V}ar, \mathcal{V}ar)$) and which ends with an existential quantifier $\exists$, the very operator Q_n cannot be ignored in the calculation of the characteristic number, as it is usually done. The reason for that is that model chekcing φ' can be done in $\mathbf{P}$ (cf. Corollary 7) and this does not allow to combine the existential quantifier of the last strategic term with another one. This is reflected in the third case of the definition of *ql*.

Definition 42 (Level i Formula) *We say that φ is a* level i formula *iff* $ql(\mathcal{UO}(\varphi)) = i$.

Model Checking $\mathcal{L}^k_{ATLP}$: Upper Bounds

Plausibility terms are quite important for the base language $\mathcal{L}^{\text{base}}_{ATLP}$; it does not make much sense to consider the logic without them. In fact, when $\mathcal{L}^{\text{base}}_{ATLP}$ formulae are considered in the context of pure CGS's, the whole logic degenerates to pure ATL. This observation does not apply to higher levels of ATLP any more. Indeed, all characterizations of game theoretic solutions concepts that we have presented are expressed as $\mathcal{L}^1_{ATLP}$ formulae *without* hard-wired terms. Moreover – as we will see – not using hard-wired terms yields an improved model checking complexity.

Below we state the main results of this section. The intuition is the following. For each level i formula we have i quantifier alternations; in addition to that, in each level there can be two more implicit quantifiers due to the cooperation modalities (*there is* a plausible strategy of the proponents such that *for all* plausible strategies of the opponents ...). It must also be ensured that the quantifiers of two nested levels are separated from each other, otherwise they can be combined; the term $\max\{0, k-i-1\}$ accounts for that.

Theorem 25 (Model Checking $\mathcal{L}^k_{ATLP}$ in Pure CGS) *For $k \geq 1$, $i \geq 0$ let φ be a level-i formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\text{gt}, \Pi, \emptyset)$. Moreover, let M be a CGS, and q a state in M. Then, model checking $M, q \models \varphi$ can be done in time $\mathbf{\Delta^P_{i+2k+1-\max\{0,k-i-1\}}}$.*

Proof in Appendix 5.11.2.

Note, that the restriction to pure CGS is essential because defining a given set of strategies Υ might require checking whether a strategy is plausible in the final nesting stage. And that case the advantage of not having hard-wired plausibility terms would vanish and the complexity would increase. So, if plausibility terms are available the last level of an ATLP formula cannot be verified in polynomial time anymore (according to Corollary 7). The complexity can increase as shown in the following result.

Theorem 26 (Model Checking $\mathcal{L}^k_{ATLP}$ in Well-Behaved CGSP) *Let φ be a level-i formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$, M a well-behaved CGSP, and q a state in M. Model checking $M, q \models \varphi$ can be done in $\mathbf{\Delta^P_{i+2(k+1)+1-\max\{0,k-i\}}}$.*

Proof in Appendix 5.11.2.

Model Checking $\mathcal{L}^k_{ATLP}$: Hardness and Completeness

As it turns out, model checking $\mathcal{L}_{ATLP}$, and even each $\mathcal{L}^k_{ATLP}$ for $k \geq 1$ is in general **PSPACE**-complete. To show the lower bounds for $\mathcal{L}^k_{ATLP}$ (with arbitrary $k \geq 1$) we show that $\mathcal{L}^1_{ATLP}$ is **PSPACE**-hard, implying that all logics $\mathcal{L}^k_{ATLP}$ (for $k \geq 1$) are **PSPACE**-hard too. That the general model checking problem for $\mathcal{L}_{ATLP}$ formulae is in **PSPACE** follows directly from the algorithm shown in Figure 5.8.

The hardness proof, similar to the one for $\mathcal{L}_{ATLP^{ATLI}}$ is rather technical and can be found in Appendix 5.11.2. As a corollary of Proposition 68, we get the following.

Theorem 27 ($\mathcal{L}^k_{ATLP}$ is PSPACE-complete) *The model checking problems for $\mathcal{L}_{ATLP}$ and for $\mathcal{L}^k_{ATLP}$ (for each $k \geq 1$) are **PSPACE**-complete.*

Proof Easiness is immediate since the model checking algorithm presented in Figure 5.8 can be executed in polynomial space with respect to the input (cf. Theorem 25 and Proposition 61). Hardness is shown by the polynomial space reduction from QSAT (Proposition 68). ■

Finally, we turn to classes in which the number of alternations is restricted by a fixed upper bound, and we conjecture that the model checking problem for i-level formulae of $\mathcal{L}^k_{ATLP}$ is in fact complete in its complexity classes determined in Theorems 25 and 26.

Conjecture 1 *Let φ be a level-i formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \emptyset)$, $k \geq 1$, $i \geq 0$. Moreover, let M be a CGS, and q a state in M. Then, model checking $M, q \models \varphi$ is $\mathbf{\Delta^P_{i+2k+1-\max\{0,k-i-1\}}}$-complete.*

Conjecture 2 *Let φ be a level-i formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$, M a well-behaved CGSP, and q a state in M. Model checking $M, q \models \varphi$ is $\mathbf{\Delta^P_{i+2(k+1)+1-\max\{0,k-i\}}}$-complete.*

5.6.4 Summary of Complexity Results

Throughout Section 5.6, we have analyzed the model checking complexity of $\mathcal{L}_{ATLP}$. The base language was shown to lie in $\mathbf{\Delta^P_3}$ with both abstract and ATLI-based plausibility terms. We also proved that model checking both logics is complete regarding this class. The complexity of model checking $\mathcal{L}^k_{ATLP}$ formulae was shown to depend on three factors:

1. The *nesting level* k of plausibility terms;
2. the *quantifier level*; and
3. whether abstract plausibility terms were present or not.

	0	1	...	i	...	unbounded
$\mathcal{L}^{\text{basic}}_{ATLP}$	$\mathbf{P}$	-	-	-	...	-
$\mathcal{L}^{0}_{ATLP}$	$\mathbf{P}$	-	...	-	...	-
$\mathcal{L}^{1}_{ATLP}$	$\mathbf{\Delta^P_3}$	$\mathbf{\Delta^P_4}$	...	$\mathbf{\Delta^P_{i+3}}$	...	**PSPACE**
$\mathcal{L}^{2}_{ATLP}$	$\mathbf{\Delta^P_4}$	$\mathbf{\Delta^P_6}$	...	$\mathbf{\Delta^P_{5+i-\max\{0,1-i\}}}$	...	**PSPACE**
$\vdots$					...	$\vdots$
$\mathcal{L}^{k}_{ATLP}$ $i > k+1$	$\mathbf{\Delta^P_{k+2}}$	$\mathbf{\Delta^P_{k+4}}$	...	$\mathbf{\Delta^P_{i+2k+1-\max\{0,k-i-1\}}}$	...	**PSPACE**

Figure 5.10: Summary of the model checking results for pure concurrent game structures (i.e., without hard-wired plausibility terms). All $\mathbf{P}$, $\mathbf{\Delta^P_3}$, and **PSPACE** results are completeness results.

	0	1	...	i	...	unbounded
$\mathcal{L}^{\text{basic}}_{ATLP}$	$\mathbf{\Delta^P_3}$	-	...	-	...	-
$\mathcal{L}^{0}_{ATLP}$	$\mathbf{\Delta^P_3}$	-	...	-	...	-
$\mathcal{L}^{1}_{ATLP}$	$\mathbf{\Delta^P_4}$	$\mathbf{\Delta^P_6}$	...	$\mathbf{\Delta^P_{i+5-\max\{0,1-i\}}}$	...	**PSPACE**
$\mathcal{L}^{2}_{ATLP}$	$\mathbf{\Delta^P_5}$	$\mathbf{\Delta^P_7}$	...	$\mathbf{\Delta^P_{7+i-\max\{0,2-i\}}}$	...	**PSPACE**
$\vdots$						$\vdots$
$\mathcal{L}^{k}_{ATLP}$ $i > k$	$\mathbf{\Delta^P_{k+3}}$	$\mathbf{\Delta^P_{k+5}}$	...	$\mathbf{\Delta^P_{i+2(k+1)+1-\max\{0,k-i\}}}$	...	**PSPACE**

Figure 5.11: Summary of the model checking results in well-behaved CGSP's. All $\mathbf{\Delta^P_3}$ and **PSPACE** results are completeness results.

The quantifier level is influenced by the number of alternations and with which quantifiers – existential or universal – sequences start and end. In general, an i-level $\mathcal{L}^{k}_{ATLP}$ formula without plausibility terms was shown to be in

$$\mathbf{\Delta^P_{i+2k+1-\max\{0,k-i-1\}}}$$

where its counterpart with hard-wired terms was marginally more difficult to check:

$$\mathbf{\Delta^P_{i+2(k+1)+1-\max\{0,k-i\}}}.$$

The results for formulae without (resp. with) primitive plausibility terms are summarized in Figure 5.10 (resp. Figure 5.11).

Note that all our game theoretic characterizations could already be expressed by $\mathcal{L}^{1}_{ATLP}$ formulae without hard-wired terms.

5.7 Conclusions

We proposed a logic in which one can study the outcome of rational play in a logical framework, under various rationality criteria. Although solving game-like scenarios with help of various solution concepts is arguably the main application of game theory, to our knowledge, there has been very little work on this issue. We are *not* discussing the merits of one rationality criterion or the other, nor the pragmatics of using particular criteria to predict the actual behaviour of agents. Our aim was to propose a *conceptual tool* in which the consequences of accepting one or another criterion can be studied.

We believe that the logic we propose provides much flexibility and modeling power. The results presented in Sections 5.5 and 5.6 also suggest that the expressive power of the language is quite high. Our main technical results are as follows:

ATLP: The very definition of the logic ATLP in Section 5.4 and the study of its expressive power in Section 5.5.1.

Classical Solution Concepts: There are several *classical* solution concepts for extensive games: Nash equilibrium, subgame perfect Nash equilibrium, undominated strategies, and Pareto optimality. We show, by *relating models of our logic (CGSP's) to extensive form games*, that these solution concepts can be formulated as formulae in ATLP (in fact, already in $\mathcal{L}^1_{ATLP}$). This is shown in Section 5.5.2

General Solution Concepts: While the classical solution concepts for games are formulated using *payoffs* (which was the reason to extend models by additional propositions), we propose to formulate *generalized solution concepts* as formulae in our logic ATLP. More precisely, we propose to use $\mathcal{L}_{ATL}$-path formulae η_i as *winning conditions* for agent i. Thus, instead of computing payoffs in an extensive form game, we consider CGSP models plus a vector of $\mathcal{L}_{ATL}$-path formulae η_i (representing the payoff for agent i). We demonstrate $\mathcal{L}^1_{ATLP}$ formulae that correctly express in ATLP our generalized solution concepts. This is elaborated in Section 5.3.5.

Model Checking in ATLP: An extensive study of the model checking complexity in several classes of models and variants of the language is presented in Section 5.6. On the way, we also define another interesting variant of ATL (where both proponents and opponents are required to use only uniform strategies) and we establish its model checking complexity.

Our ultimate goal is to come up with a logic that would allow us to study strategies, time, knowledge, and plausible/rational behaviour under both perfect and imperfect information. However, putting so many dimensions in one framework at once is usually not a good idea – even more so in this case because the interaction between abilities and knowledge is non-trivial (cf. [83, 75, 63]). In [25], we have investigated *time, knowledge and plausibility*. In this article, we studied *strategies, time and rationality*. We hope to integrate both views into a single powerful framework in the future.

We would like to thank two anonymous referees for pointing out several issues that helped us to improve (and shorten) this article.

5.8 Appendix: Bargaining with Discount

In Example 22 we presented bargaining with discount. After each round the worth of the goods is reduced by δ_i. In round t the goods have a value of $r(\delta_i^t)$. Because we use a rounding function r, there is a minimal round T such that $r(\delta_i^{T+1}) = 0$ for $i = 1$ or $i = 2$. We can treat this case as finite horizon bargaining game [123, 101].

Now, consider the case that a_i's opponent, denoted by a_{-i}, is the offerer in T. It can offer 0 and a_i should accept, because in the next round the goods are worthless for a_i.

On the other hand, if a_i is offerer in T we have to distinguish two cases. If $r(\delta_{-i}^{T+1}) = 0$ then following the same reasoning as before a_i can offer 0 to a_{-i}. In the other case, namely $r(\delta_{-i}^{T+1}) \neq 0$, we consider the subsequent round $T+1$ in which a_{-i} takes the role as offerer and can successfully offer 0 to i.

Now, it is possible to solve the game starting from the end. Solutions for $\delta_1 = \delta_2$ can be found in the literature [101]. Here, we recall the idea for different discount rates.

At first, let a_1 be the last offerer and $r(\delta_2^{T+1}) = 0$. This implies, that T is even (the initial round is 0). In T, a_1 offers $\langle 1, 0\rangle$ and a_2 accepts. Knowing this, in $T-1$ agent a_2 can offer $\langle \delta_1, 1-\delta_1\rangle$, since in the next round the value of the good for a_1 would become reduced by δ_1. Following the same reasoning, in $T-2$ a_1 could successfully offer $\langle 1-\delta_2(1-\delta_1), \delta_2(1-\delta_1)\rangle$. Finally, in round $t = 0$ a_1 can offer $\langle \zeta, 1-\zeta\rangle$ where

$$\zeta := (1-\delta_2)\sum_{i=0}^{\frac{T}{2}-1}(\delta_1\delta_2)^i + (\delta_1\delta_2)^{\frac{T}{2}} = (1-\delta_2)\frac{1-(\delta_1\delta_2)^{\frac{T}{2}}}{1-\delta_1\delta_2} + (\delta_1\delta_2)^{\frac{T}{2}}$$

Secondly, consider the case in which a_2 is the last offerer in T and $r(\delta_1^{T+1}) = 0$. This time T is odd but the reasoning stays the same. In round 0 a_1 can offer $\langle \zeta', 1-\zeta'\rangle$ where

$$\zeta' := (1-\delta_2)\frac{1-(\delta_1\delta_2)^{\frac{T+1}{2}}}{1-\delta_1\delta_2}$$

5.9 Appendix: Uniform ATL$_{ir}$

In this section, we introduce and investigate the logic of "uniform ATL$_{ir}$" (ATL$^u_{ir}$). We use the logic only for technical reasons, namely it provides the intermediate step in the completeness proof for the complexity of model checking ATLP. Still, we believe that the logic can be interesting in itself. Moreover, the technique we use for proving the completeness is interesting too (and gives insight into the complexity as well as the relationship between the problem we study and known complexity from game theory).

The idea is based on Schobbens's ATL$_{ir}$ [121], i.e., ATL for agents with imperfect information and imperfect recall. There, it was assumed that the

coalition A in formula $\langle\langle A\rangle\rangle_{ir}\varphi$ can only use strategies that assign same choices in indistinguishable states (so called *uniform* strategies). Then, the outcome of every strategy of A was evaluated in every possible behaviour of the remaining agents $\mathbb{A}\mathrm{gt} \setminus A$ (with no additional assumption with respect to that behaviour). In ATL$^u_{ir}$, we assume that the opponents ($\mathbb{A}\mathrm{gt} \setminus A$) are also required to respond *with a uniform memoryless strategy*. The syntax of ATL$^u_{ir}$ is the same as that of ATL, only cooperation modalities are annotated with additional tags ir and u to indicate the imperfect **i**nformation and **r**ecall, and **u**niformity of all agents' strategies.

5.9.1 Semantics

The semantics of ATL$^u_{ir}$ can be defined as follows. Firstly, we define models as *concurrent epistemic game structures* (CEGS), i.e. CGS with epistemic relations $\sim_a \subseteq St \times St$, one per agent. (The intended meaning of $q \sim_a q'$ is that agent a cannot distinguish between between states q and q'.) Secondly, we require that agents have the same options in indistinguishable states, i.e., that $q \sim_a q'$ implies $d_a(q) = d_a(q')$. A (memoryless) strategy s_A is *uniform* if $q \sim_a q'$ implies $s^a_A(q) = s^a_A(q')$ for all $q, q' \in St, a \in A$. To simplify the notation, we define $[q]_a = \{q' \mid q \sim_a q'\}$ to be the class of states indistinguishable from q for a; $[q]_A = \bigcup_{a\in A}[q]_a$ collects all the states that are indistinguishable from q for some member of the group A; finally, $out(Q, s_A) = \bigcup_{q\in Q} out(q, s_A)$ collects all the execution paths of strategy s_A from states in set Q.

Now, the semantics is given by the clauses below:

$M, q \models p$ iff $p \in \pi(q)$

$M, q \models \neg\varphi$ iff $M, q \not\models \varphi$

$M, q \models \varphi \wedge \psi$ iff $M, q \models \varphi$ and $M, q \models \psi$

$M, q \models \langle\langle A\rangle\rangle^u\bigcirc\varphi$ iff there is a uniform strategy s_A such that, for every uniform counterstrategy $t_{\mathbb{A}\mathrm{gt}\setminus A}$, and $\lambda \in out([q]_A, \langle s_A, t_{\mathbb{A}\mathrm{gt}\setminus A}\rangle)$,[15] we have $M, \lambda[1] \models \varphi$;

$M, q \models \langle\langle A\rangle\rangle^u\Box\varphi$ iff there is a uniform strategy s_A such that, for every uniform counterstrategy $t_{\mathbb{A}\mathrm{gt}\setminus A}$, and $\lambda \in out([q]_A, \langle s_A, t_{\mathbb{A}\mathrm{gt}\setminus A}\rangle)$, we have $M, \lambda[i] \models \varphi$ for all $i = 0, 1, ...$;

$M, q \models \langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\psi$ iff there is a uniform strategy s_A such that, for every uniform counterstrategy $t_{\mathbb{A}\mathrm{gt}\setminus A}$, and $\lambda \in out([q]_A, \langle s_A, t_{\mathbb{A}\mathrm{gt}\setminus A}\rangle)$, there is $i \in \mathbb{N}_0$ with $M, \lambda[i] \models \psi$, and $M, \lambda[j] \models \varphi$ for all $0 \leq j < i$.

5.9.2 Model Checking Complexity

We show the lower bound by reduction of SNSAT$_2$, a typical $\mathbf{\Delta^P_3}$-complete problem. We recall the definition of SNSAT$_i$ after [96].

[15] Note that the definition of concurrent game structures, that we use after [8], implies that CGS are deterministic, so there is in fact exactly one such path λ.

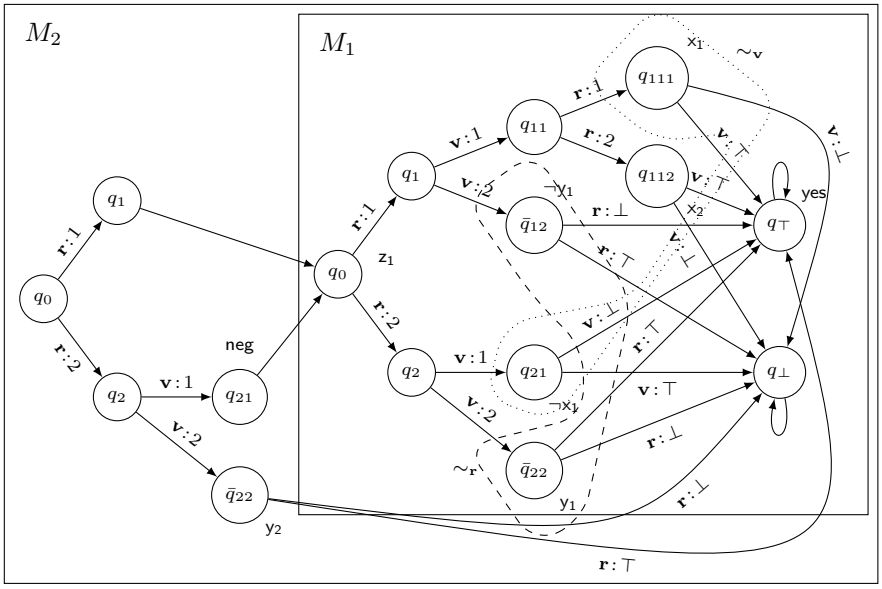


Figure 5.12: CEGS M_2 for $\varphi_1 \equiv ((x_1 \wedge x_2) \vee \neg y_1) \wedge (\neg x_1 \vee y_1)$, $\varphi_2 \equiv z_1 \wedge (\neg z_1 \vee y_2)$.

Definition 43 (SNSAT$_i$)
Input: *p sets of propositional variables $X_r^j = \{x_{1,r}^j, ..., x_{k,r}^j\}$ for each $j = 1, \ldots, i$; p propositional variables z_r, and p Boolean formulae φ_r in positive normal form (i.e., negation is allowed only on the level of literals). Each φ_r involves only variables in $\bigcup_{j=1}^{i} X_r^j \cup \{z_1, ..., z_{r-1}\}$, with the following requirement:*
$z_r \equiv \exists X_r^1 \forall X_r^2 \exists X_r^3 \ldots Q X_r^i . \varphi_r(z_1, ..., z_{r-1}, X_r^1, \ldots, X_r^i)$, where $Q = \forall$ (resp. $Q = \exists$) if i is even (resp. odd).
Output: *The value of z_p.*

In this section we focus on SNSAT$_2$ where we set $X_r^1 = X_r = \{x_{1,r}, ..., x_{k,r}\}$ and $X_r^2 = Y_r = \{y_{1,r}, ..., y_{k,r}\}$.

Our reduction of SNSAT$_2$ is an extension of the reduction of SNSAT presented in [81, 82]. That is, we construct the CEGS M_r corresponding to z_r with two players: *verifier* $\mathbf{v}$ and *refuter* $\mathbf{r}$. The CEGS is turn-based, that is, every state is "governed" by a single player who determines the next transition. Each subformula $\chi_{i_1 \ldots i_l}$ of φ_r has a corresponding state $q_{i_1 \ldots i_l}$ in M_r. If the outermost logical connective of φ_r is $\wedge$, the refuter decides at q_0 which subformula χ_i of φ_r is to be satisfied, by proceeding to the "subformula" state q_i corresponding to χ_i. If the outermost connective is $\vee$, the verifier decides which subformula χ_i of φ_r will be attempted at q_0. This procedure is repeated until all subformulae are single literals. The states corresponding to literals are called "proposition" states.

The difference from the construction from [81, 82] is that formulae are in positive normal form (rather than CNF) and that we have two kinds of "proposition" states now: $q_{i_1 \ldots i_l}$ refers to a literal consisting of some $x \in X_r$ and is governed by $\mathbf{v}$; $\bar{q}_{i_1 \ldots i_l}$ refers to some $y \in Y_r$ and will be governed by $\mathbf{r}$. Now, the values of the underlying propositional variables x, y are declared at the "proposition" states, and the outcome is computed. That is, if $\mathbf{v}$ executes

$\top$ for a positive literal, i.e. $\chi_{i_1\dots i_l} = x$, (or $\bot$ for $\chi_{i_1\dots i_l} = \neg x$) at $q_{i_1\dots i_l}$, then the system proceeds to the "winning" state $q_\top$; otherwise, the system goes to the "sink" state $q_\bot$. For states $\bar{q}_{i_1\dots i_l}$ the procedure is analogous. Models corresponding to subsequent z_r are nested like in Figure 5.12.[16] "Proposition" states referring to the same variable x are indistinguishable for $\mathbf{v}$ (so that he has to declare the same value of x in all of them), and the states referring to the same y are indistinguishable for $\mathbf{r}$. A sole ATL^u_{ir} proposition yes holds only in the "winning" state $q_\top$. As in [81, 82], we have the following result which concludes the reduction.

Proposition 66 *The above construction shows a polynomial reduction of SNSAT$_2$ to model checking ATL^u_{ir} in the following sense. Let*

$$\begin{aligned}\Phi_1 &\equiv \langle\!\langle \mathbf{v}\rangle\!\rangle^u_{ir}(\neg\mathsf{neg})\,\mathcal{U}\,\mathsf{yes}, \quad \textit{and}\\ \Phi_r &\equiv \langle\!\langle \mathbf{v}\rangle\!\rangle^u_{ir}(\neg\mathsf{neg})\,\mathcal{U}\,(\mathsf{yes}\vee(\mathsf{neg}\wedge\langle\!\langle\varnothing\rangle\!\rangle^u_{ir}\bigcirc\neg\Phi_{r-1})) \quad \textit{for } r=2,\dots,p.\end{aligned}$$

Then, we have z_p iff $M_p, q_0^p \models_{ATL^u_{ir}} \Phi_p$.

As for the upper bound, we note that there is a straightforward $\mathbf{\Delta_3^P}$ algorithm that model-checks formulae of ATL^u_{ir}: when checking $\langle\!\langle A\rangle\!\rangle^u_{ir}T\varphi$ in M, q, it first recursively checks φ (bottom-up), and labels the states where φ held with a special proposition yes. Then, the algorithm guesses a uniform strategy s_A and calls an oracle that guesses a uniform counterstrategy $t_{\mathbb{A}\mathrm{gt}\setminus A}$. Finally, it trims M according to $\langle s_A, t_{\mathbb{A}\mathrm{gt}\setminus A}\rangle$, and calls a CTL model checker to check formula ATyes in state q of the resulting model. This gives us the following result.

Theorem 28 *Model checking ATL^u_{ir} is $\mathbf{\Delta_3^P}$-complete with respect to the number of transitions in the model and the length of the formula. It is $\mathbf{\Delta_3^P}$-complete even for turn-based CEGS with at most two agents.*

5.10 Appendix: From ATL^u_{ir} to ATLP with ATLI-Based Plausibility Terms

The reduction of ATL^u_{ir} model checking to model checking of $ATLP^{ATLI}$ in "pure" CGS is rather sophisticated. We do not present a reduction for full model checking of ATL^u_{ir}; it is enough to show the reduction for the kind of models that we get in Appendix 5.9.2 (i.e., turn-based models with two agents, two "final" states $q_\top, q_\bot$, no cycles except for the loops at the final states, and uncertainty appearing only in states one step before the end of the game, cf. Figure 5.12).

Firstly, we reconstruct the concurrent epistemic game structure M_p from Section 5.9.2 so that the last action profile is always "remembered" in the final states. Then, we show how uniformity of strategies can be characterized with a formula of ATLI extended with epistemic operators. Thirdly, we show how the model and the formula can be transformed to get rid of epistemic links and operators (yielding a "pure" CGS and a formula of "pure" ATLI).

[16] All states in the model for z_r are additionally indexed by r.

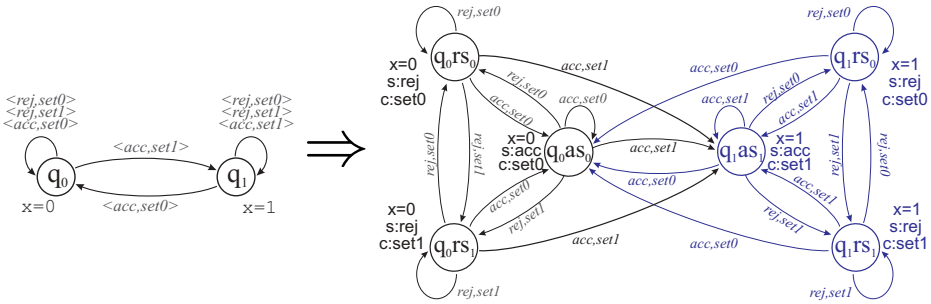


Figure 5.13: Memorizing the last action profile in a simple 2-agent system

Finally, we show how the resulting characterization of uniformity can be "plugged" into an ATLP formula to require that only uniform strategy profiles are taken into account.

Adding More Final States to the Model. To recall, the input of ATL^u_{ir} model checking consists in our case of a concurrent epistemic game structure M_p (like the one in Figure 5.12) and an ATL^u_{ir} formula Φ_p (cf. Proposition 66). We begin the reduction by reconstructing M_p to M'_p in which the last action profile is "remembered" in the final states. The idea is based on the construction from [52, Proposition 16] where it is applied to all states of the system, cf. Figure 5.13.

In our case, we first create copies of states $q_\top, q_\bot$, one per incoming transition. That is, the construction yields states of the form $\langle q, \alpha_1, \ldots, \alpha_k\rangle$, where $q \in \{q_\top, q_\bot\}$ is a final state of the original model M_p, and $\langle \alpha_1, \ldots, \alpha_k\rangle$ is the action profile executed just before the system proceeded to q. Each copy has the same valuation of propositions as the original state q, i.e., $\pi'(\langle q, \alpha_1, \ldots, \alpha_k\rangle) = \pi(q)$. Then, for each action $\alpha \in Act$ and agent $i \in \mathbb{A}\mathrm{gt}$, we add a new proposition $\mathsf{i} : \alpha$. Moreover, we fix the valuation of $\mathsf{i} : \alpha$ in M'_p so that it holds exactly in the final states that can be achieved by an action profile in which i executes α (i.e., states $\langle q, \alpha_1, ..., \alpha_i, ..., \alpha_k\rangle$). Note that the number of both states and transitions in M'_p is linear in the transitions of M_p. The transformation produces model M'_p which is equivalent to M_p in the following sense. Let φ be a formula of ATL^u_{ir} that does not involve special propositions $\mathsf{i} : \alpha$. Then, for all $q \in St$: $M_p, q \models_{ATL^u_{ir}} \varphi$ iff $M'_p, q \models_{ATL^u_{ir}} \varphi$.

In M'_p, agents can "recall" their actions executed at states that involved some uncertainty (i.e., states in which the image of some indistinguishability relation $\sim_i$ was not a singleton). Now we can use ATLI (with additional help of knowledge operators, see below) to characterize uniformity of strategies.

Characterizing Uniformity in ATLI+K. We will now show that uniformity of a strategy can be characterized in ATLI *extended with epistemic operators* K_a (that we call ATLI+K. $K_a\varphi$ reads as "agent a knows that φ". The semantics of ATLI+K extends that of ATLI by adding the standard semantic clause from epistemic logic:

$$M, q \models K_a\varphi \text{ iff } M, q' \models \varphi \text{ for every } q' \text{ such that } q \sim_a q'.$$

We note that ATLI+K can be also seen as ATEL [133] extended with intentions.

Let us now consider the following formula of ATLI +Knowledge:

$$uniform(\sigma) \equiv (\mathbf{str}\ \sigma)\langle\langle\emptyset\rangle\rangle\Box \bigwedge_{i\in\mathbb{A}\mathrm{gt}} \bigvee_{\alpha\in d(i,q)} K_i\langle\langle\emptyset\rangle\rangle\bigcirc \mathsf{i} : \alpha.$$

The reading of $uniform(\sigma)$ is: suppose that profile σ is played ($\mathbf{str}\ \sigma$); then, for all reachable states ($\langle\langle\emptyset\rangle\rangle\Box$), every agent has a single action ($\bigwedge_{i\in\mathbb{A}\mathrm{gt}} \bigvee_{\alpha\in d(i,q)}$) that is determined for execution ($\langle\langle\emptyset\rangle\rangle\bigcirc \mathsf{i} : \alpha$) in every state indistinguishable from the current state (K_i). Thus, formula $uniform(\sigma)$ characterizes the *uniformity* of strategy profile σ. Formally, for every concurrent epistemic game structure M, we have that $M, q \models_{\text{ATLI+K}} uniform(\sigma)$ iff $[\sigma[a]]$ is uniform for each agent $a \in \mathbb{A}\mathrm{gt}$ (for all states reachable from q). Of course, only reachable states matter when we look for strategies that should enforce a temporal goal.

Note that the epistemic operator K_a refers to incomplete information, but σ is now an arbitrary (i.e., not necessarily uniform) strategy profile. We observe that the length of the formula is linear in the number of agents and actions in the model.

Translating Knowledge to Ability. To get rid of the epistemic operators from formula $uniform(\sigma)$ and epistemic relations from model M'_p, we use the construction from [70] (which refines that from [52, Section 4.4]). The construction yields a concurrent game structure $tr(M'_p)$ and an ATLI formula $tr(uniform(\sigma))$ with the following characteristics. The idea can be sketched as follows. The set of agents becomes extended with *epistemic agents* e_i (one per $a_i \in \mathbb{A}\mathrm{gt}$), yielding $\mathbb{A}\mathrm{gt}'' = \mathbb{A}\mathrm{gt} \cup \mathbb{A}\mathrm{gt}^e$. Similarly, the set of states is augmented with *epistemic states* q^e for every $q \in St'$ and $e \in \mathbb{A}\mathrm{gt}^e$; the states "governed" by the epistemic agent e_a are labeled with a special proposition $\mathsf{e_a}$. The "real" states q from the original model are called "action" states, and are labeled with another special proposition act. Epistemic agent e_a can enforce transitions to states that are indistinguishable for agent a (see Figure 5.14 for an example).[17] Then, "a knows φ" can be rephrased as "e_a can only effect transitions to epistemic states where φ holds". With some additional tricks to ensure the right interplay between actions of epistemic agents, we get the following translation of formulae:

$$\begin{aligned}
tr(p) &= p, \qquad \text{for } p \in \Pi \\
tr(\neg\varphi) &= \neg tr(\varphi) \\
tr(\varphi \vee \psi) &= tr(\varphi) \vee tr(\psi) \\
tr(\langle\langle A\rangle\rangle\bigcirc\varphi) &= \langle\langle A \cup \mathbb{A}\mathrm{gt}^e\rangle\rangle\bigcirc(\mathsf{act} \wedge tr(\varphi)) \\
tr(\langle\langle A\rangle\rangle\Box\varphi) &= \langle\langle A \cup \mathbb{A}\mathrm{gt}^e\rangle\rangle\Box(\mathsf{act} \wedge tr(\varphi)) \\
tr(\langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\psi) &= \langle\langle A \cup \mathbb{A}\mathrm{gt}^e\rangle\rangle(\mathsf{act} \wedge tr(\varphi))\,\mathcal{U}\,(\mathsf{act} \wedge tr(\psi)) \\
tr(K_i\varphi) &= \neg\langle\langle e_1, ..., e_i\rangle\rangle\bigcirc\big(\mathsf{e_i} \wedge \langle\langle e_1, ..., e_k\rangle\rangle\bigcirc(\mathsf{act} \wedge \neg tr(\varphi))\big).
\end{aligned}$$

[17] The interested reader is referred to [70] for the technical details of the construction.

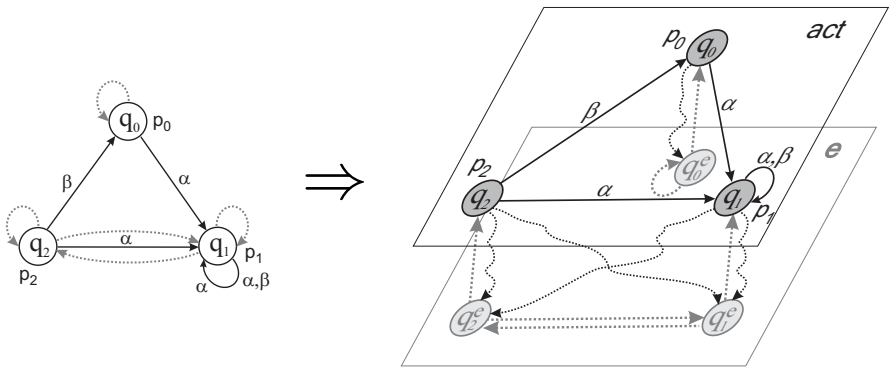


Figure 5.14: Getting rid of knowledge and epistemic links

Note that the length of $tr(\varphi)$ is linear in the length of φ and the number of agents k. Two important facts follow from [70, Theorem 8]:

Lemma 13 *For every CEGS M and a formula of ATL^u_{ir} that does not include the special propositions* $\mathsf{act}, \mathsf{e}_1, \ldots, \mathsf{e}_k$*, we have* $M, q \models_{ATL^u_{ir}} \varphi$ *iff* $tr(M), q \models_{ATL^u_{ir}} tr(\varphi)$.

Lemma 14 *For every CEGS M, we have* $M, q \models_{ATLI+K} uniform(\sigma)$ *iff* $tr(M), q \models_{ATLI+K} tr(uniform(\sigma))$.

Putting the Pieces Together: the Reduction. We observe that ATL^u_{ir} can be seen as ATL where only uniform strategy profiles are allowed. An ATLI formula that characterizes uniformity has been defined in the previous paragraphs. It can be now plugged into our "ATL with Plausibility" to restrict agents' behaviour in the way the semantics of ATL^u_{ir} does. This way, we obtain a reduction of $SNSAT_2$ to model checking of $ATLP^{ATLI}$.

Proposition 67
z_p *iff* $tr(M'_p), q^p_0 \models_{ATLP^{ATLI}} (\textbf{\textit{set-pl}}\ \sigma.tr(uniform(\sigma)))\mathbf{Pl}\, tr(\Phi_p)$.

Proof We have z_p iff $M'_p, q^p_0 \models_{\mathrm{ATL}^u_{ir}} \Phi_p$ iff $tr(M'_p), q^p_0 \models_{\mathrm{ATL}^u_{ir}} tr(\Phi_p)$ iff $tr(M'_p), q^p_0 \models_{ATLP^{ATLI}} (\textbf{set-pl}\ \sigma.tr(uniform(\sigma)))\mathbf{Pl}\, tr(\Phi_p)$. ∎

5.11 Appendix: Some Model Checking Complexity Proofs

5.11.1 Results in Section 5.6.1

Proposition 59: Let M be a well-behaved CGSP, q a state in M, and φ a formula of $\mathcal{L}^{\text{base}}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$. Then $M, q \models \varphi$ iff $mcheckATLP(M, q, \varphi)$. The algorithm runs in time $\mathbf{\Delta^P_3}$ with respect to the number of transitions in the model and the length of the formula.

Proof Function $mcheck$ is called recursively, at most l times. All cases apart from $\varphi \equiv \langle\langle A\rangle\rangle \bigcirc \psi$ where ψ includes no $\langle\langle C\rangle\rangle$ (analogously for the other temporal operators) can be performed in polynomial time. Now, there is a nondeterministic Turing machine A_B which implements function $beatable$: Firstly, it guesses a strategy t possibly together with another witness necessary for $plausiblestrat$ (by assumption the latter is in $\mathbf{NP}$) and verifies if t is plausible, the verification can be done in polynomial time (by the same assumption). Finally, if t is plausible A_B has to perform CTL model checking which lies in $\mathbf{P}$.

It remains to show that there is a nondeterministic oracle Turing machine A_S with oracle A_B implementing $solve$. (Formally, the machine requires two oracles, one answering the question whether s is plausible, and the other is given by A_B. However, the former is computationally less expensive then the latter and can be ignored since we are interested in the oracle with the highest complexity.) A_S works as follows: Firstly, it guesses a profile s (again possibly together with a witness for $plausiblestrat$); secondly, it verifies whether s is plausible and then calls oracle A_B and inverts its answer. Altogether, there are polynomial many calls to machine $A_S^{A_B} \in \mathbf{NP}^{\mathbf{NP}}$. This renders the algorithm to be in $\mathbf{\Delta_3^P}$. ∎

Proposition 62: Let $c \in \mathbb{N}$ be a constant. Model checking $\mathcal{L}^{\text{base}}_{ATLP}$ formulae with respect to the class of well-behaved bounded models $\mathfrak{M}^c$ can be done in polynomial time with respect to the number of transitions in the model and the length of the formula.

Proof sketch We modify the original ATL model checking procedure as follows. Consider the formula $\varphi \equiv \langle\langle A\rangle\rangle\gamma$ where γ is a pure ATL path formula. Let B be the set of agents assumed to play plausibly and let $\Upsilon \neq \Sigma$ be the current set of plausible strategies described by some term and state. For each $s_B \in \Upsilon|_B$ we remove from M all transitions which cannot occur according to s_B, yielding model M^{s_B}, and check whether $M^{s_B}, q \models_{ATL} \langle\langle A\rangle\rangle\gamma$. We proceed like this for all $s \in \Upsilon|_B$ (there are only polynomially many). This procedure is incorporated into our ATLP model checking algorithm and applied bottom up. ∎

5.11.2 Results in Section 5.6.3

Upper Bounds

First, we recall a basic complexity result that will be used in the rest of this section. Then, we present proofs of upper bounds for model checking $\mathcal{L}^k_{ATLP}$ for pure CGS's and well-behaved CGSP's.

Remark 24 *A relation $R \subseteq \times_{i=1}^{k+1}\Sigma^*$ ($k \geq 1$) is called* polynomial decidable *whenever there is a deterministic Turing machine (DTM) which decides $\{(x, y_1 \ldots, y_k) : (x, y_1 \ldots, y_k) \in R\}$ in polynomial time; furthermore, R is called* polynomial balanced *if there is a $k \in \mathbb{N}$ such that for all $(x, y_1 \ldots, y_k) \in R$: $|y_i| \leq |x|^k$ for all $i = 1, \ldots k$.*

For a language L and $k \geq 1$ the following holds: $L \in \Sigma_k^P$ if, and only if, there is a polynomial decidable and balanced $(k+1)$-ary relation R such that $L = \{x \mid$

$\exists y_1 \forall y_2 \exists y_3 \ldots Q y_k\ ((x, y_1 \ldots, y_k) \in R)\}$ *where* $Q = \forall$ *(resp.* $Q = \exists$*) if* k *is odd (resp.* k *even) [112, Corollary 2 of Theorem 17.8].*

Theorem 25: Let φ be a level-i formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \emptyset)$, $k \geq 1$, $i \geq 0$. Moreover, let M be a CGS, and q a state in M. Then, model checking $M, q \models \varphi$ can be done in time $\mathbf{\Delta^P_{i+2k+1-\max\{0,k-i-1\}}}$.

Proof By induction over k. In the following we restrict ourselves to (**set-pl** $\cdot$) without loss of generality.

Case $k = 1$**.** Let φ be a level-i $\mathcal{L}^1_{ATLP}$ formula, (**set-pl** ω) an operator occurring in φ such that $l(\{(\textbf{set-pl } \omega)\}) = i$ and $\omega = \sigma.Q_1\sigma_1 Q_2\sigma_2 \ldots Q_n\sigma_n\varphi'$ where

$$\varphi' \in \mathcal{L}^{\text{base}}_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \{\sigma, \sigma_1, \ldots, \sigma_n\}).$$

Note that $M^{s,s_1,\ldots,s_n}, q \models \varphi'$ can be checked in polynomial time since all constructible plausibility terms are rectangular and the representation is directly given (see Corollary 7). Moreover, let q' denote the state in which ω is evaluated. W.l.o.g. we can assume that φ has the following structure:

$$\varphi \equiv (\textbf{set-pl } \omega)\mathbf{Pl}\, \langle\langle A \rangle\rangle \Box \mathsf{yes}$$

Now, φ is true in M and q if and only if there is a plausible strategy s_A for A and *no* plausible strategy t with $t|_A = s$ such that $M', q \models_C TL\neg\mathsf{A}\Box\mathsf{yes}$ where M' is the trimmed model of M wrt t. In the following we neglect the complexity needed to verify whether s_A is plausible since the method $beatable$ also verifies this property and its complexity is as least as high (cf. proof of Proposition 59). Thus, φ is true if, and only if:

$\exists s_A \neg\big(\exists t\ (t \in \widehat{[\![\omega]\!]}^{q'}$ and $R_{\models}(M, q, s_A, t, \Box\mathsf{yes}))\big)$ iff
$\exists s_A \neg\ \big(\exists t Q_1 s_1 Q_2 s_2 \ldots Q_n s_n\ (M^{t,s_1,\ldots,s_n}, q' \models \varphi'$ and $R_{\models}(M, q, s_A, t, \Box\mathsf{yes}))\big)$
iff $\exists s_A \forall t \bar{Q}_1 s_1 \bar{Q}_2 s_2 \ldots \bar{Q}_n s_n\ (M^{t,s_1,\ldots,s_n}, q' \not\models \varphi'$ or $\neg R_{\models}(M, q, s_A, t, \Box\mathsf{yes}))$

where $R_{\models}(M, q, s_A, t, \Box\mathsf{yes}) = true$ iff $t|_A = s_A$ and $M', q \models_C TL\neg\mathsf{A}\Box\mathsf{yes}$ where M' is the "trimmed" model of M wrt t, and $\bar{Q}$ is the dual operator to Q.

Now, the latter conditions can be verified in polynomial time. We consider the number of quantifier alternations. Subsequent strategies which are quantified by quantifiers of the same type can be guessed together. The same holds if the sequence starts with existential quantifiers. These strategies can be guessed together with strategy t. A quantifier level of $l(\{(\textbf{set-pl } \omega)\}) = i$ denotes that it is sufficient to alternatingly guess i witnesses. We obtain the following structure:

$$\exists s_A \forall x_t \exists x_1 \forall x_2 \ldots Q x_i$$

where $Q = \exists$ (resp. $Q = \forall$) if i is even (resp. odd). Where x_i denotes a witness for a strategy or several strategies if guessing can be combined.

Thus, according to Remark 24 checking whether φ is satisfied can be determined in time Σ_{i+2} and the complete model checking algorithm for level-i $\mathcal{L}^1_{ATLP}$ formula can be performed in time $\mathbf{\Delta^P_{i+3}}$ (there can be polynomial many such constructs).

Induction step: $k \mapsto k+1$ **($k > 1$).** Let φ be a level-i $\mathcal{L}^{k+1}_{ATLP}$ formula and letω be a term in φ of the form $\omega = \sigma_1.Q_1\sigma_1Q_2\sigma_2 \ldots Q_n\sigma_n\varphi'$ such that $l((\textbf{set-pl}\ \omega)) = i$. Furthermore, let $\text{RALT}(Q_1 \ldots Q_n) = j$; then, $l_{\varphi'} := ql(\mathcal{UO}(\varphi')) = i - j$ and φ' is an $\mathcal{L}^k_{ATLP}$ formula. Thus, by induction hypothesis we have that φ' can be model checked in time

$$\mathbf{\Delta^P_{r+1}} \quad \text{where } r := l_{\varphi'} + 2k - \max\{0, k - l_{\varphi'} - 1\}.$$

Again, w.l.o.g. we can assume that φ has the following structure:

$$\varphi \equiv (\textbf{set-pl}\ \omega)\mathbf{Pl}\,\langle\langle A\rangle\rangle\Box\mathsf{yes}.$$

We proceed as in case $k = 1$. Firstly, a profile s is guessed, then a profile t and it is checked whether t is plausible and coincides with s wrt A and whether the trimmed model (wrt t) satisfies $\neg\mathsf{A}\Box\mathsf{yes}$. We obtain the following structure:

$$\exists s_A \neg \left(\exists t \, (t \in \widehat{[\![\omega]\!]}^{q'} \text{ and } R_{\models}(M, q, s_A, t, \Box\mathsf{yes})) \right) \quad \text{iff}$$

$$\exists s_A \neg \left(\exists t Q_1 s_1 Q_2 s_2 \ldots Q_n s_n \, (\underbrace{M^{t,s_1,\ldots,s_n}, q' \models \varphi'}_{\in \mathbf{\Delta^P_{r+1}}} \text{ and } \underbrace{R_{\models}(M, q, s_A, t, \Box\mathsf{yes})}_{\in \mathbf{P}}) \right)$$

Since $M^{t,s_1,\ldots,s_n}, q' \models \varphi'$ is invoked by a nondeterministic polynomial Turing machine we can assume that its model checking problem can be solved in $\mathbf{\Sigma^P_r}$ instead of $\mathbf{\Delta^P_{r+1}}$; the polynomial effort of the deterministic machine can also be done by the invoking nondeterministic machine. Hence to verify $M^{t,s_1,\ldots,s_n}, q' \models \varphi'$ witnesses according to

$$\exists x_1 \forall x_2 \exists x_3 \ldots Q_r x_r$$

have to be guessed; then, the question whether φ is satisfied with respect to the witnesses $x_1, \ldots, x_r$ can be solved in polynomial time.

Because $\text{RALT}(Q_1Q_2 \ldots Q_n) = j$ it suffices to guess witnesses according to the following structure:

$$\forall x'_1 \exists x'_2 \forall x'_3 \ldots \forall x'_j.$$

If $Q_1Q_2 \ldots Q_n$ would start (resp. end) with existential quantifiers the corresponding witnesses could be guessed together with the one for profile t (resp. witness x_1). Putting things together the following witnesses have to be guessed:

$$(\star) \qquad \exists s_A \forall x_t \, \exists x'_1 \forall x'_2 \exists x'_3 \ldots \exists x'_j \, \forall x_1 \exists x_2 \forall x_3 \ldots \bar{Q}_r x_r$$

It remains to show that the number of alternations in $(\star)$ does never exceed $i + 2(k+1) - \max\{0, (k+1) - i - 1\}$

We distinguish two cases $(k+1) - i - 1 \leq 0$ and $(k+1) - i - 1 > 0$.

Case: $k + 1 - i - 1 \leq 0$**.** That is, $k \leq i$. We are going to determine the maximal possible number of alternations in $(\star)$.

Firstly, assume that $j \geq 1$. That is the number of alternation is given by $2 + j + r = i + 2(k+1) - \max\{0, k - i + j - 1\}$. This expression is maximal whenever $k - i + j - 1 \leq 0$. Because of $k \leq i$ this is always the case for $j = 1$. In this case the formula has at most

$$i + 2(k+1) - \max\{k + 1 - i - 1\} \quad \text{alternations.}$$

For $j = 0$ there is at least one alternation less, since the witness x_t can be guessed together with x_1.

Case: $k + 1 - i - 1 > 0$**.** That is, $k > i$. Firstly, we consider the case $j \geq 1$. There are at most $i+2(k+1)-\max\{0, k-i+j-1\}$ alternations, where the number becomes maximal for $j = 1$; i.e. we have at most

$$i + 2(k+1) - \max\{k + 1 - i - 1\} \quad \text{alternations.}$$

Now, we consider the case $j = 0$. In this case there are at most $i+2(k+1)-1-\max\{0, k-i-1\}$ alternations. Because of $k > i$, we have that $k - i - 1 \geq 0$ and, hence $i+2(k+1)-1-\max\{0, k-i-1\}$ is equivalent to $i + 2(k+1) - \max\{0, (k+1) - i - 1\}$.

Thus, $i + 2(k+1) - \max\{k+1-i-1\}$ alternations denotes the maximal possible number of alternations which proofs our claim the model checking algorithm for level-i $\mathcal{L}^{k+1}_{ATLP}$ can be performed in time $\mathbf{P}^{\mathbf{\Sigma^P_{i+2(k+1)-\max\{k+1-i-1\}}}} = \mathbf{\Delta^P_{i+2(k+1)+1-\max\{k+1-i-1\}}}$.

■

Theorem 26: Let φ be a level-i formula of $\mathcal{L}^k_{ATLP}(\mathbb{A}\mathrm{gt}, \Pi, \Omega)$, M a well-behaved CGSP, and q a state in M. Model checking $M, q \models \varphi$ is in $\mathbf{\Delta^P_{i+2(k+1)+1-\max\{0,k-i\}}}$.

Proof The proof is similar to the one of Theorem 25. In comparison to the claim of Theorem 25, $2k$ has changed to $2(k+1)$ and $\max\{0, k-i-1\}$ to $\max\{0, k-i\}$. The reason for this is that the final nesting (i.e. formulae in $\mathcal{L}^{\text{base}}_{ATLP}$) might contain hard-wired terms and it can not be verified in polynomial time anymore. This causes the change from k to $k+1$ (it requires to guess s_A and verify it against all responses t). However, now the complexity might be increased too much since the final strategy s_A of A could be guessed together with the next to last strategy t' of the opponents ($\exists s'_A \neg(\exists t' \exists s_A \neg(\exists t))$) if there is no further alternation between t' and s_A, caused by a plausibility term. Such an "interfering" alternation is only possible if the given formula is at least an level-k formula; this is reflected by $\max\{0, k-i\}$. ■

PSPACE-completeness of $\mathcal{L}^k_{ATLP}$ Model Checking

We use *quantified satisfiability* (QSAT) to show **PSPACE**-completeness of model checking $\mathcal{L}^k_{ATLP}$ and $\mathcal{L}_{ATLP}$.

Definition 44 (QSAT [112])
Input: *A boolean formula φ in conjunctive normal with i variables $x_1, \ldots, x_i$.*
Output: *True if $\exists x_1 \forall x_2 \ldots Q_i x_i \; \varphi$ is satisfiable, false otherwise (where $Q = \forall$ if i is even, and $Q = \exists$ if i is odd).*

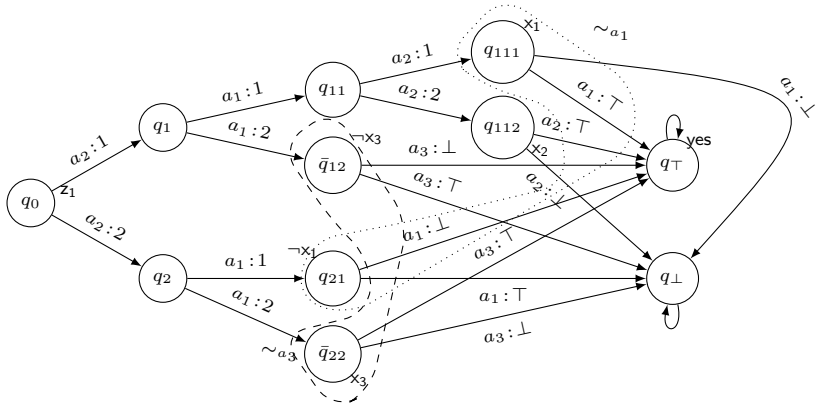


Figure 5.15: Construction of the intermediate model M'_φ for $\varphi \equiv \exists x_1 \forall x_2 \exists x_3((x_1 \wedge x_2) \vee \neg x_3) \wedge (\neg x_1 \vee x_3)$.

Given an instance φ of QSAT we construct an $\mathcal{L}^1_{ATLP}$ formula θ_φ and a CGSP M_φ (both are constructible in polynomial space regarding the length of φ) such that φ is satisfiable if, and only if, $M_\varphi, q_0 \models \theta_\varphi$. In the following we sketch the constructions which are based on the reduction of SNSAT$_2$ to model checking ATL$^u_{ir}$ proposed in Appendix 5.9.2, and the translation of ATL^u_{ir} to $\mathcal{L}_{ATLP^{ATLI}}$ proposed in Appendix 5.10.

Let $\varphi \equiv \exists x_1 \forall x_2 \ldots Q_n x_n \; \psi$ be an instance of QSAT. Firstly, we sketch the construction of the CEGS M'_φ which will then be transformed into a CGSP M_φ. In comparison to the construction in Appendix 5.9.2, we consider n agents one for each quantifier (in fact, we consider $\max\{2, n\}$ agents; however, for the rest of this section we assume that $n \geq 2$). The agent belonging to quantifier i is named a_i. Except for the proposition states the procedure is completely analogous to the construction given in Appendix 5.9.2 where agent a_2 is considered as *refuter* and a_1 as *verifier*. (Alternatively, two additional agents could be added.) The procedure at the proposition states changes as follows: In such a state, say q, referring to a literal l, say $l = x_i$, agent a_i can decide on the value of x_i. Note again that the agent is required to make the same choice in indistinguishable states. In Figure 5.15 the construction is shown for the formula $\varphi \equiv \exists x_1 \forall x_2 \exists x_3((x_1 \wedge x_2) \vee \neg x_3) \wedge (\neg x_1 \vee x_3)$. Finally, the model M_φ is obtained from M'_φ by following the same steps as described in Appendix 5.9.2.

Secondly, we construct formula θ_φ from φ as follows:

$$\theta_\varphi \equiv (\textbf{set-pl}\ \sigma_1 . \forall \sigma_2 \exists \sigma_3 \ldots Q_n \sigma_n \chi) \mathbf{Pl} \langle\langle \mathbb{A}\mathrm{gt} \rangle\rangle \bigcirc \top$$

where

$$\chi \equiv \left(\bigwedge_{i=1,\ldots,n} \mathit{uniform}^i_A TLP(\sigma_i) \right) \wedge (\textbf{set-pl}\ \langle \sigma_1[1], \ldots, \sigma_n[n] \rangle) \mathbf{Pl} \langle\langle \emptyset \rangle\rangle \Diamond \mathsf{yes}.$$

Next, we will give the intuition behind θ_φ. Firstly, it is easy to see that $\mathbf{Pl} \langle\langle \mathbb{A}\mathrm{gt} \rangle\rangle \bigcirc \top$ is true whenever the set of plausible strategy profiles is *not empty*.

Hence, the actual set of strategies described by the preceding (**set-pl** $\cdot$) operator is not particularly important, rather if *some* strategy is plausible or not.

Secondly, note that (**set-pl** $\langle \sigma_1[1], \ldots, \sigma_n[n] \rangle$) in χ describes a single strategy profile and that all individual strategies can be considered independently (the set is rectangular, cf. Definition 41 and Lemma 11). Furthermore, an individual strategy is mainly used to assign $\top$ or $\bot$ to propositional variables in the proposition states. (Except for agents a_1 and a_2 which also take on the refuter and verifier role; they can also perform actions in non-proposition states.) Hence, a given strategy profile can be seen as a valuation of the propositional variables.

Thirdly, we analyze χ with respect to a given profile $\sigma := \langle \sigma_1[1], \ldots, \sigma_n[n] \rangle$ taking into account the previous points. By formula $uniform^i_A TLP(\sigma_i)$ it is ensured that agent i assigns the same valuation to propositions in indistinguishable states. Now, χ is true if the "winning state" $q_\top$ is reached by following the strategy described by σ (it describes a unique path in the model). In other words, χ is true if, and only if, the valuation described by σ satisfies φ.

Finally, due to the previous observations, if $[\![\sigma_1.\forall\sigma_2\widehat{\exists\sigma_3 \ldots Q_n\sigma_n\chi}]\!]$ is non-empty it can be interpreted as follows: There is a valuation of x_1 such that for all valuations of x_2 there is a valuation of x_3, and so forth such that φ is satisfied.

The following proposition states that the construction is correct.

Proposition 68 *Let φ be a QSAT instance. Then it holds that φ is satisfiable if, and only if, $M_\varphi, q_0 \models \theta_\varphi$ where M_φ and θ_φ are effectively constructible from φ in polynomial space with respect to the length of the formula φ.*

Proof sketch Let φ be a QSAT instance. We use the construction above to obtain M'_φ and θ_φ where $uniform^i_A TLP(\sigma)$ is obtained as follows: Firstly, we take the ATLI +K formula $uniform(\sigma|_i)$ (where $\sigma|_i$ refers to agent i's startegy in σ) as described in Appendix 5.10; then, we use the polynomial translation to change knowledge to ability, yielding a pure ATLI formula. Finally, we use the polynomial translation from $ATLI$ to $ATLP$ given in Section 5.5.1 (Proof of Proposition 49) to obtain a pure ATLP formula $uniform^i_A TLP(\sigma)$. Hence, the latter formula is true if agent i's strategy contained in the complete profile σ is a uniform strategy. This shows that θ_φ can be constructed in polynomial space.

Model M_φ is obtained from M'_φ by the same scheme. Firstly, the construction from [70] referred to in Appendix 5.10 is applied. Secondly, the resulting CGS with intentions is transformed to a CGSP using the construction from Section 5.5.1 (Proposition 49) again. The constructed model M_φ is also polynomial with respect to φ.

We get that φ is satisfiable if, and only if, $M_\varphi, q_0 \models \theta_\varphi$. ∎

Acknowledgments

This article is dedicated to Victor W. Marek on the occasion of his 65. birthday. Jŕgen Dix very gratefully acknowledges Victor's help and support in the past.

Part III

Verification in Multi-Agent Systems

Chapter 6

Model Checking Abilities of Agents: A Closer Look (joint work with Jürgen Dix)

Abstract. Alternating-time temporal logic (ATL) is a logic for reasoning about open computational systems and multi-agent systems. It is well known that ATL model checking is *linear in the size of the model*. We point out, however, that the size of an ATL model is usually *exponential in the number of agents*. When the size of models is defined in terms of *states and agents* rather than *transitions*, it turns out that the problem is (1) $\mathbf{\Delta_3^P}$-complete for concurrent game structures, and (2) $\mathbf{\Delta_2^P}$-complete for alternating transition systems. Moreover, for "Positive ATL" that allows for negation only on the level of propositions, model checking is (1) $\mathbf{\Sigma_2^P}$-complete for concurrent game structures, and (2) $\mathbf{NP}$-complete for alternating transition systems. We show a nondeterministic polynomial reduction from checking arbitrary alternating transition systems to checking turn-based transition systems, We also discuss the determinism assumption in alternating transition systems, and show that it can be easily removed.
In the second part of the paper, we study the model checking complexity for formulae of *ATL with imperfect information* (ATL_{ir}). We show that the problem is $\mathbf{\Delta_2^P}$-complete in the number of transitions and the length of the formula (thereby closing a gap in previous work of Schobbens [121]). Then, we take a closer look and use the same fine structure complexity measure as we did for ATL with perfect information. We get the surprising result that checking formulae of ATL_{ir} is also $\mathbf{\Delta_3^P}$-complete in the general case, and $\mathbf{\Sigma_2^P}$-complete for "Positive ATL_{ir}". Thus, model checking agents' abilities for both perfect and imperfect information systems belongs to the same complexity class when a finer-grained analysis is used.

`Keywords:` multi-agent systems, model checking, computational complexity.

6.1 Introduction

Alternating-time temporal logic [6, 7, 8] is one of the most interesting frameworks that emerged recently for reasoning about computational systems. One of the most appreciated features of ATL is its model checking complexity: *linear in the size of the model* (more precisely, in the *number of transitions* in the model) *and the size of the formula*. Thus, the complexity is the same as for computation tree logic CTL (despite ATL being strictly more expressive than CTL). While the result is certainly attractive, it guarantees less than one could expect. We point out that, unlike in CTL, the number of transitions outgoing from a single global state in an ATL model is usually *exponential* in the number of agents. While it is well-known that the number of states in a model can be exponential in the size of a higher-level description of the system, it also turns out that the size of an *explicit* ATL model is usually exponential in the number of agents, *even when no higher level description is considered*.

Following this observation, we study the precise ATL model checking complexity for *explicit models* when *the size of models is defined in terms of states* rather than transitions, and *the number of agents is considered a parameter of the problem*. In fact, we show that the model checking problem is intractable in such a setting of input parameters. Firstly, it turns out that the problem is $\mathbf{\Sigma_2^P}$-complete for the language of "Positive ATL" (where negation is allowed only on the level of propositions) and the semantics based on concurrent game structures (CGS). Secondly, model checking "Positive ATL" is "only" $\mathbf{NP}$-complete when an earlier semantics, based on alternating transition systems (ATS), is used. Using our ideas, Laroussinie et al. [95] have proved that model checking formulae of the full ATL is $\mathbf{\Delta_3^P}$-complete for CGS, and $\mathbf{\Delta_2^P}$-complete for ATS; we cite their results, too.[1]

Additionally, we point out that ATL model checking over the broader class of nondeterministic alternating transition systems is still $\mathbf{\Delta_2^P}$-complete for the full language, and $\mathbf{NP}$-complete for the "positive" sublanguage—which suggests that using the more general class of ATS may be a good choice in practice.

The results mentioned above apply to general, arbitrary models (be it CGS or ATS). On the other hand, model checking ATL for turn-based models (i.e., those in which only one agent/process at a time is executing an action) can still be done in deterministic polynomial time. We show how, for an arbitrary alternating transition system M, a turn-based system M' can be constructed, so that a combination of *choices* in M corresponds to a combination of *strategies* in a fragment of M'. We then propose a translation of ATL formulae into ATL$^+$ formulae, such that the original formula holds in M, q iff the translated formula holds in M', q. Finally, we point out that the latter can be model-checked in time $\mathbf{\Delta_2^P}$ (respectively $\mathbf{NP}$ for "Positive ATL"), and

[1] A note of explanation. $\mathbf{\Delta_2^P} = \mathbf{P^{NP}}$ is the class of problems that can be solved by a *deterministic* Turing machine in polynomial time with adaptive calls to an $\mathbf{NP}$ oracle. $\mathbf{\Sigma_2^P} = \mathbf{NP^{NP}}$ is the class of problems that can be solved by a *nondeterministic* Turing machine in polynomial time with calls to an $\mathbf{NP}$ oracle. $\mathbf{\Delta_3^P} = \mathbf{P^{\Sigma_2^P}}$ is the class of problems that can be solved by a *deterministic* Turing machine in polynomial time with adaptive calls to a $\mathbf{\Sigma_2^P}$ oracle [112, 13].

Contrary to what the index suggests, $\mathbf{\Delta_{i+1}^P}$ belongs still to the i-th level of the polynomial hierarchy.

thus provide another (slightly more general) proof that the original problem is in $\mathbf{\Delta_3^P}$ (resp. $\mathbf{\Sigma_2^P}$ for "Positive ATL"). The translation of models is independent from the translation of formulae in our construction, which allows for "pre-compiling" models when one wants to check various properties of a particular multi-agent system.

The last part of the paper is concerned with *ATL with imperfect information* (ATL$_{ir}$), introduced by Schobbens in [121]. Since no satisfying semantics based on alternating transition systems for strategic abilities under imperfect information has been proposed so far, we present our results for an epistemic extension of concurrent game structures only. First, we close a gap in Schobbens's results, and show that model checking ATL$_{ir}$ formulae is $\mathbf{\Delta_2^P}$-complete with respect to the number of transitions in the model and the length of the formula (thus confirming the initial intuition of Schobbens [121]). We also show that the problem is $\mathbf{NP}$-complete for "Positive ATL$_{ir}$". Next, we demonstrate that model checking ATL$_{ir}$ is $\mathbf{\Delta_3^P}$-complete when the size of models is defined in terms of states and agents rather than transitions (and $\mathbf{\Sigma_2^P}$-complete for "Positive ATL$_{ir}$" in the same setting). We point out that the result is somewhat surprising: *checking abilities of agents acting under imperfect information falls into the same complexity class as checking abilities of agents in perfect information scenarios* when a finer-grained analysis is used.

This article is organised as follows. In Section 6.2 we introduce ATL and its semantics. Several variants of this logic are considered and the notions of *perfect* and *imperfect* information in these systems are precisely defined. Section 6.3 presents known and new results on the complexity of model checking ATL with concurrent game structures. In Section 6.4 we consider model checking ATL with alternating transition systems. We also show that the usual *singleton* requirement in ATS can be relaxed without affecting the complexity. In Section 6.5 we relate general ATS and turn-based systems. Section 6.6 contains our main results with respect to agents with imperfect information. They suggest, rather surprisingly, that there is no major difference in the model checking complexity between games of perfect and imperfect information. We conclude with Section 6.7.

6.1.1 History of the Results

This article is based on preliminary results presented in a series of conference papers [78, 79, 80]. In [78] we considered model checking of ATL, observing (correctly) that the "perceived" size of ATL models is very sensitive to the measure one applies (much more so than CTL models). We concluded that the model checking problem was $\mathbf{\Sigma_2^P}$-complete for CGS, and $\mathbf{NP}$-complete for ATS. Unfortunately, *these* claims were incorrect, as Laroussinie, Markey and Oreiby pointed out in [95]. The error was related to the way we handled negation in our model checking algorithms. Laroussinie and colleagues used our ideas to obtain the *correct* results, namely to prove that ATL model checking is $\mathbf{\Delta_3^P}$-complete for concurrent game structures, and $\mathbf{\Delta_2^P}$-complete for alternating transition systems [95]. Still, they observed that our algorithms *were* correct for "Positive ATL" – i.e., ATL without negated cooperation modalities. We summarize all the relevant results in Section 6.3 of this paper, to get the complete picture.

Another paper [79], where we reported complexity results on model checking ATL_{ir}, suffered from the error mentioned above. Again, our claims were correct for "Positive ATL_{ir}", but incorrect for model checking of the full logic. In Section 6.6.2, we present an entirely new result, proving that model checking of full ATL_{ir} is $\mathbf{\Delta_2^P}$-hard (and hence, by Schobbens's result [121], also $\mathbf{\Delta_2^P}$-complete) with respect to the number of transitions in the model. Then we use the results, obtained by Laroussinie et al. for ATL [95], to establish the precise model checking complexity of ATL_{ir} with respect to the number of states and agents.

6.2 ATL: A Logic of Strategic Ability

The logic of ATL [6, 7, 8] was originally invented to capture properties of *open computer systems* (such as computer networks), where different components can act autonomously, and computations in such systems result from their combined actions. Alternatively, ATL can be seen as a logic for systems involving multiple agents, that allows one to reason about what agents can achieve in game-like scenarios. ATL can be also understood as a generalisation of the well-known branching time logic CTL [39, 38], in which path quantifiers E ("there is a path") and A ("for each path") are replaced by *cooperation modalities* $\langle\!\langle A \rangle\!\rangle$ that express strategic abilities of agents and their teams.

Formula $\langle\!\langle A \rangle\!\rangle \varphi$ expresses that agents A have a collective strategy to enforce φ. ATL formulae include temporal operators: "$\bigcirc$" ("in the next state"), $\Box$ ("always from now on") and $\mathcal{U}$ ("until"). An additional operator $\Diamond$ ("sometime from now on") can be defined as $\Diamond \varphi \equiv \top \mathcal{U} \varphi$. Like in CTL, every occurrence of a temporal operator is preceded by exactly one cooperation modality (this variant of the language is sometimes called "vanilla" ATL). The broader language of ATL*, in which no such restriction is imposed, is discussed briefly in Section 6.2.3.

Formally, the recursive definition of ATL formulae is:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\!\langle A \rangle\!\rangle \bigcirc \varphi \mid \langle\!\langle A \rangle\!\rangle \Box \varphi \mid \langle\!\langle A \rangle\!\rangle \varphi \mathcal{U} \varphi.$$

"Positive ATL" is the subset of ATL in which the use of negation is limited to propositional formulae. Formally, it can be defined by the following grammar:

$$\varphi ::= p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A \rangle\!\rangle \bigcirc \varphi \mid \langle\!\langle A \rangle\!\rangle \Box \varphi \mid \langle\!\langle A \rangle\!\rangle \varphi \mathcal{U} \varphi.$$

A number of different semantics and model classes have been defined for ATL, most of them equivalent (cf. [51, 52]). Among these, *concurrent game structures* [8] are probably the most natural and easiest to come up with when modeling concrete problem domains. Moreover, they are the easiest to extend to the imperfect information case, because actions have global identity in concurrent game structures (cf. [66]). However, it seems that *alternating transition systems*, introduced in the more preliminary papers [6, 7] may offer some advantage in terms of model checking complexity (see the results in Sections 6.3 and 6.4).

In what follows, we begin with a brief presentation of the two most prominent semantics, based on concurrent game structures and alternating transition systems. In Section 6.2.4, we are extending the scope of ATL with the possibility that some agents have imperfect information about the current state of the world. Research on this subject is far from being complete, yet a number of ATL extensions have already been proposed to cope with such systems: from the logics of ATEL [133, 134] and "ATL with incomplete information" [8] to more sophisticated approaches like ATOL and ATEL-R* [83], ATL$_{ir}$ and ATL$_{iR}$ [121], ETSL [135], and CSL [73]. Among these, ATL$_{ir}$ seems to stand out for its simplicity and conceptual clarity; also (unlike for logics of agents with perfect recall, e.g. ATEL-R* and ATL$_{iR}$), its model checking procedure is decidable. We believe that ATL$_{ir}$, while probably *not* the definitive ATL extension for games with imperfect information,[2] includes constructs that are indispensable when addressing such games. Thus, we treat ATL$_{ir}$ as a kind of "core" ATL-based language for strategic ability under imperfect information, and present its syntax and semantics in Section 6.2.4.

6.2.1 Strategic Abilities with Concurrent Game Structures

Concurrent game structures (CGS) [8], can be defined as tuples

$$M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o \rangle,$$

where:

- $\mathbb{A}\mathrm{gt} = \{a_1, ..., a_k\}$ is a finite nonempty set of all agents,
- St is a nonempty set of states,
- Π is a set of atomic propositions,
- $\pi : \Pi \to 2^{St}$ is a valuation of propositions,
- Act is a finite nonempty set of (atomic) actions;
- function $d : \mathbb{A}\mathrm{gt} \times St \to 2^{Act}$ defines actions available to an agent in a state, and
- o is a (deterministic) transition function that assigns outcome states $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to states and tuples of actions.

Remark 25 *Firstly, this variant of concurrent game structures differs slightly from the original CGS [8]: we represent agents and their actions with symbolic labels, whereas in [8] they are represented with natural numbers.*

Secondly, determinism is not a crucial issue here, as systems with nondeterministic outcome of agents' actions can be modeled easily by introducing a new, fictional agent, "nature", which settles all nondeterministic transitions.

[2] ATOL, for example, is strictly more expressive with the same model checking complexity.

A *strategy* of agent a is a conditional plan that specifies what a is going to do in each possible state. Thus, a strategy can be represented with a function $s_a : St \to Act$, such that $s_a(q) \in d_a(q)$. A *collective strategy* for a group of agents $A = \{a_1, ..., a_r\}$ is simply a tuple of strategies $S_A = \langle s_{a_1}, ..., s_{a_r} \rangle$, one per agent from A.

Remark 26 *This is an important deviation from the original semantics of ATL [6, 7, 8], where strategies assign agents' choices to* sequences *of states, which suggests that agents can recall the whole history of each game. In this article, however, we employ "memoryless" strategies. While the choice of one or another notion of strategy affects the semantics of the full ATL*, and most ATL extensions (e.g. for games with imperfect information), it should be pointed out that both types of strategies yield equivalent semantics for "pure" ATL [121].*

A *path* in M is an infinite sequence of states that can result from subsequent transitions, and refers to a possible course of action (or a possible computation). Function $out(q, S_A)$ returns the set of all paths that may occur when agents A execute strategy S_A from state q onward:[3]

$out(q, S_A) = \{\lambda = q_0 q_1 q_2 ... \mid q_0 = q$ and for each $i = 1, 2, ...$ there exists a tuple of agents' decisions $\langle \alpha^{i-1}_{a_1}, ..., \alpha^{i-1}_{a_k} \rangle$ st. $\alpha^{i-1}_a \in d_a(q_{i-1})$ for every $a \in \mathbb{A}\mathrm{gt}$, and $\alpha^{i-1}_a \in S_A(a)(q_{i-1})$ for every $a \in A$, and $o(q_{i-1}, \alpha^{i-1}_{a_1}, ..., \alpha^{i-1}_{a_k}) = q_i\}$.

Let $\lambda[i]$ denote the ith position in computation λ (starting from $i = 0$). The semantics of ATL is defined via the clauses below. Informally speaking, $M, q \models \langle\langle A \rangle\rangle \Phi$ iff there exists a collective strategy S_A such that Φ holds for all computations from $out(q, S_A)$.

$M, q \models p$ iff $q \in \pi(p)$ (where $p \in \Pi$);

$M, q \models \neg\varphi$ iff $M, q \not\models \varphi$;

$M, q \models \varphi \vee \psi$ iff $M, q \models \varphi$ or $M, q \models \psi$;

$M, q \models \langle\langle A \rangle\rangle \bigcirc \varphi$ iff there is a collective strategy S_A such that, for each path $\lambda \in out(S_A, q)$, we have $M, \lambda[1] \models \varphi$;

$M, q \models \langle\langle A \rangle\rangle \Box \varphi$ iff there exists S_A such that, for each $\lambda \in out(S_A, q)$, we have $M, \lambda[i] \models \varphi$ for every $i \geq 0$;

$M, q \models \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$ iff there exists S_A such that, for each $\lambda \in out(S_A, q)$, there is $i \geq 0$ for which $M, \lambda[i] \models \psi$, and $M, \lambda[j] \models \varphi$ for each $0 \leq j < i$.

Example 33 *Consider a modified version of the Simple Rocket Domain from [18]. There is a rocket that can be moved between London (*roL*) and Paris (*roP*), and piece of cargo that can lie in London (*caL*), Paris (*caP*), or inside the rocket (*caR*). Three agents are involved:* 1 *who can load the cargo, unload it, or move the rocket;* 2 *who can unload the cargo or move the rocket, and* 3 *who can load the cargo or supply the rocket with fuel. Each agent can also stay idle at a particular moment (the nop – "no-operation" actions). The "moving" action has the highest priority. "Loading"*

[3] The notation $S_A(a)$ stands for the strategy s_a of agent a in the tuple $S_A = \langle s_{a_1}, ..., s_{a_r} \rangle$.

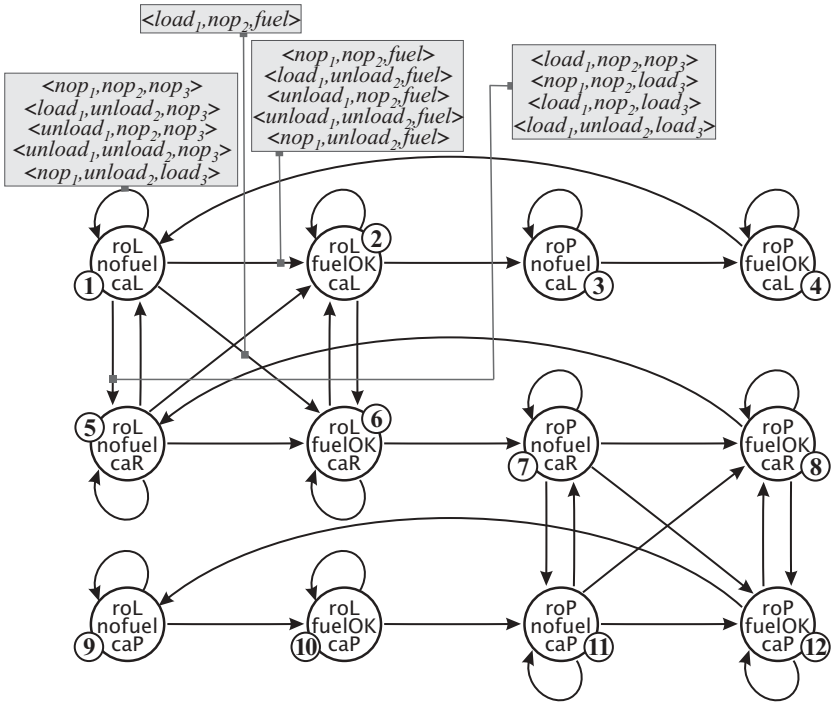


Figure 6.1: A CGS for Simple Rocket Domain

is executed when the rocket does not move and more agents try to load than to unload; "unloading" works in a similar way (in a sense, the agents "vote" whether the cargo should be loaded or unloaded). Finally, "fueling" can be accomplished only when the rocket tank is empty (alone or in parallel with loading or unloading). The rocket can move only if it has some fuel (fuelOK*), and the fuel must be refilled after each flight. The concurrent game structure for the domain is shown in Figure 6.1 (we will refer to this model as* M_1*). All the transitions for state* 1 *(the cargo and the rocket are in London, no fuel in the rocket) are labeled; output of agents' choices for other states is analogous.*

Example ATL formulae that hold in $M_1, 1$ *are:* $\neg\langle\langle 1\rangle\rangle\Diamond\mathsf{caP}$ *(agent* 1 *cannot deliver the cargo to Paris on his own),* $\langle\langle 1,3\rangle\rangle\Diamond\mathsf{caP}$ *(*1 *and* 3 *can deliver the cargo if they cooperate), and* $\langle\langle 2,3\rangle\rangle\Box(\mathsf{roL} \wedge \langle\langle 2,3\rangle\rangle\Diamond\mathsf{roP})$ *(*2 *and* 3 *can keep the rocket in London forever, and at the same time retain the ability to change their strategy and move the rocket to Paris).*

It is worth pointing out that the CTL path quantifiers A and E can be embedded in ATL in the following way: $\mathsf{A}\varphi \equiv \langle\langle\varnothing\rangle\rangle\varphi$ and $\mathsf{E}\varphi \equiv \langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\varphi$. Note that the determinism of the transition function is essential for the latter property. In a deterministic system, a collective strategy for the "grand coalition" of agents $\mathbb{A}\mathrm{gt}$ determines a *single* path in the model. In contrast, this is usually not the case in non-deterministic systems. Thus, it may be the case that there is a single path for which property φ holds (i.e., we have $\mathsf{E}\varphi$), and yet the agents are not able to enforce it, so $\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\varphi$ does not hold.

On the other hand, $\mathsf{A}\varphi$ *is* equivalent to $\langle\langle\varnothing\rangle\rangle\varphi$ even when we abandon the determinism assumption (to see this, it is sufficient to check what the semantic clauses for $\langle\langle\varnothing\rangle\rangle\bigcirc\varphi$, $\langle\langle\varnothing\rangle\rangle\Box\varphi$ and $\langle\langle\varnothing\rangle\rangle\varphi\,\mathcal{U}\,\psi$ look like). This allows us to define $\mathsf{E}\bigcirc\varphi$ as $\neg\langle\langle\varnothing\rangle\rangle\bigcirc\neg\varphi$, $\mathsf{E}\Box\varphi$ as $\neg\langle\langle\varnothing\rangle\rangle\Diamond\neg\varphi$, and $\mathsf{E}\Diamond\varphi$ as $\neg\langle\langle\varnothing\rangle\rangle\Box\neg\varphi$. Still, as demonstrated in [94], $\mathsf{E}\varphi\,\mathcal{U}\,\psi$ cannot be expressed with any combination of the above operators.[4]

6.2.2 Semantics of ATL Based on ATS

Previous versions of ATL were defined over alternating transition systems [6, 7]. An *alternating transition system* (ATS) is a tuple

$$M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, \delta\rangle,$$

where:

- $\mathbb{A}\mathrm{gt}$ is a non-empty finite set of *agents*, St is a non-empty set of *states*, Π is a set of (atomic) *propositions*, and $\pi : St \to 2^{\Pi}$ is a *valuation* of propositions;
- $\delta : St \times \mathbb{A}\mathrm{gt} \to 2^{2^{St}}$ is a function that maps pairs $\langle state, agent\rangle$ to non-empty families of choices with respect to possible next states. The idea is that, at state q, agent a chooses a set $Q_a \in \delta(q,a)$ thus forcing the outcome state to be from Q_a. The resulting transition leads to a state which is in the intersection of all Q_a for $a \in \mathbb{A}\mathrm{gt}$ and so it reflects the will of all agents. Since the system is required to be deterministic (given the state and the agents' decisions), $Q_{a_1} \cap \ldots \cap Q_{a_k}$ must always be a singleton.

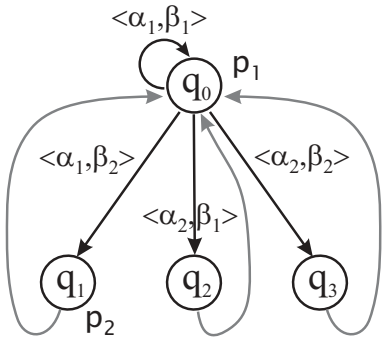
In an ATS, the type of a strategy function is slightly different since choices are sets of states now, and a strategy is represented as a mapping $s_a : St \to 2^{St}$, such that $s_a(q) \in \delta(q,a)$. The rest of the semantics looks exactly the same as for concurrent game structures. In particular, the semantic clauses are exactly the same as the ones in Section 6.2.1.

Example 34 *Consider ATS M_2 from Figure 6.2. We use symbols α_1, α_2 and β_1, β_2 as shorthands for the choices, to make the example easier to read. The following ATL formulae hold in M_2, q_0: $\neg\langle\langle a\rangle\rangle\Diamond\mathsf{p}_2$ (a cannot enforce that p_2 is eventually true), $\langle\langle a,b\rangle\rangle\Box\mathsf{p}_1$ (a and b can cooperate to guarantee that p_1 always holds), and $\langle\langle a\rangle\rangle\bigcirc(\mathsf{p}_1 \vee \mathsf{p}_2)$ (a can achieve $\mathsf{p}_1 \vee \mathsf{p}_2$ in the next step).*

Note that M_2 is not "tight" in the sense that some choices include states that cannot be reached via these choices. It can be tightened by removing q_1 from the choices in $\delta(q_1,a)$, $\delta(q_2,b)$ and $\delta(q_3,b)$, which yields an equivalent tight *ATS. We discuss the notion of tightness in Section 6.4.1 more formally.*

It is worth pointing out that alternating transition systems are usually less natural and more difficult to come up with than concurrent game structures; they are also larger in most cases (cf. [67], Section 2.7.4). More precisely: for each ATS there exists an isomorphic CGS, but the reverse does not hold. Moreover, alternating transition systems do not allow easily for extensions (e.g. with the possibility that agents may have imperfect information). This subject is discussed in more detail in [66, 52, 67].

[4] See also Remark 28, and the expressivity results from [95].



$\mathbb{A}\mathrm{gt} = \{a, b\}$
$St = \{q_0, q_1, q_2, q_3\}$
$\Pi = \{p_1, p_2\}$
$\pi(p_1) = \{q_0\}, \pi(p_2) = \{q_1\}$

$$\delta(q_0, a) = \{\{q_0, q_1\}, \{q_2, q_3\}\} \quad (\alpha_1, \alpha_2)$$
$$\delta(q_0, b) = \{\{q_0, q_2\}, \{q_1, q_3\}\} \quad (\beta_1, \beta_2)$$

$$\delta(q_1, b) = \delta(q_2, a) = \delta(q_3, a) = \{\{q_0\}\}$$
$$\delta(q_1, a) = \delta(q_2, b) = \delta(q_3, b) = \{\{q_0, q_1\}\}$$

Figure 6.2: Alternating transition system M_2: two agents, each has two choices at state q_0

Remark 27 *Note that the determinism assumption* is *significant in the case of ATS. Unlike for CGS, adding an auxiliary player ("nature") to an existing alternating transition system is neither easy nor straightforward. The problem is to extend the existing choice function δ so that it still satisfies the rigid formal requirement that* all *the intersections of choices are singletons. Designing a completely new ATS from scratch is probably an easier solution.*

We note here that model checking of ATL formulae has been proved to be linear in the size of the model and the length of the formula for both concurrent game structures [8] and alternating transition systems [7], which coincides with the model checking complexity for CTL [30]. We will discuss this issue in more detail in Section 6.3.1.

6.2.3 Beyond ATL: ATL$^+$ and ATL*

The full language of ATL* [7, 8] is usually presented as consisting of

1. *state formulae* $\langle\langle A\rangle\rangle\varphi$, expressing strategic abilities of agents to enforce specific paths of computation, and
2. *path formulae* $\bigcirc\varphi$ and $\varphi\,\mathcal{U}\,\psi$, expressing temporal properties of paths.

Both state and path formulae can be combined using Boolean operators. State formulae are interpreted in states, with $M, q \models \langle\langle A\rangle\rangle\varphi$ meaning "there is S_A such that, for each path $\lambda \in out(q, S_A)$, we have $M, \lambda \models \varphi$". Path formulae are interpreted in paths, with $M, \lambda \models \bigcirc\varphi$ and $M, \lambda \models \varphi\,\mathcal{U}\,\psi$ defined in the obvious way.

ATL* is more costly in computational terms. Model checking ATL* with memoryless strategies (i.e., the variant that we are interested in here) is **PSPACE**-complete [121]. Model checking ATL* with perfect recall is even more expensive: it is **2EXPTIME**-complete in the number of transitions in the model and the length of the formula [8].

In this article, we are only interested in its subset ATL$^+$ [121], in which each temporal operator is preceded by a single cooperation modality, modulo Boolean operators. That is, $\langle\langle A\rangle\rangle$ is followed by a Boolean combination

of path formulae $\bigcirc\varphi$, $\varphi\,\mathcal{U}\,\psi$, in which φ, ψ are state formulae again. As an example, the following is an ATL$^+$ formula: $\langle\langle a\rangle\rangle(\Box(\mathsf{p}_1 \vee \mathsf{p}_2) \wedge \Diamond\mathsf{p}_1)$. It states that a has a strategy to visit state q_0 at least once, while staying in states q_0, q_1 all the time (note, by the way, that the formula holds in M_2, q_0 from Example 34).

ATL$^+$ can be seen as a generalisation of CTL$^+$ [40]. Model checking of ATL$^+$ has been proved Δ_3-complete in the number of transitions and the length of the formula (for both memoryless and perfect recall strategies) [121], while CTL$^+$ model checking is Δ_2-complete [96]. However, the ATL$^+$ and CTL$^+$ formulae that we use in this article can be model checked in nondeterministic polynomial time (cf. Section 6.5.2).

6.2.4 Strategic Abilities under Imperfect Information

ATL and its models include no way of addressing uncertainty that an agent or a process may have about the current situation; moreover, strategies in ATL can define different choices for any pair of different states, hence implying that an agent can recognise each (global) state of the system, and act accordingly. Thus, it can be argued that the logic is tailored for describing and analyzing systems in which every agent/process has *complete and accurate knowledge* about the current state of the system. This is usually not the case for most application domains, where a process can access its *local* state, but the state of the environment and the (local) states of other agents can be observed only partially.

One of the main challenges, when a logic of strategic abilities under imperfect information is addressed, is the question of how agents' knowledge should interfere with the agents' available strategies. When reasoning about what an agent can *enforce*, it seems more appropriate to require the agent to know his winning strategy rather than to know only that such a strategy exists [66, 83, 86]. This problem is closely related to the distinction between knowledge *de re* and knowledge *de dicto*, well known in the philosophy of language [117], as well as in research on the interaction between knowledge and action [105, 106, 143]. Several variations on "ATL with imperfect information" have been proposed [83, 121, 86, 135, 73], yet none of them has been commonly accepted. In this article, we treat Schobbens' ATL$_{ir}$ and ATL$_{iR}$ [121] as "core", minimal ATL-based languages for strategic ability under imperfect information. The first logic enables reasoning about agents that have no implicit memory of the game (i.e., they use "memoryless" strategies), while the latter is guided by the assumption that agents can always memorise the whole game. As agents seldom have unlimited memory, and logics of strategic ability with imperfect information and perfect recall are believed to have undecidable model checking, we use ATL$_{ir}$ as *the* logic of strategic ability under uncertainty here.

ATL$_{ir}$ includes the same formulae as ATL, only the cooperation modalities are presented with a subscript: $\langle\langle A\rangle\rangle_{ir}$ to indicate that they address agents with imperfect *information* and imperfect *recall*. Like for ATL, *"Positive ATL$_{ir}$"* is the subset of ATL$_{ir}$ in which the use of negation is limited to propositional formulae. Models of ATL$_{ir}$, *imperfect information concurrent game structures* (i-CGS), can be presented as concurrent game structures augmented with a

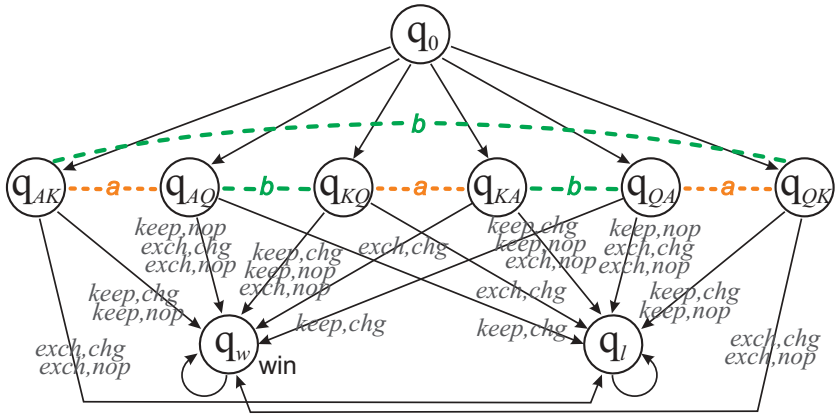


Figure 6.3: Gambling Robots game

family of indistinguishability relations $\sim_a \subseteq St \times St$, one per agent $a \in \mathbb{A}\mathrm{gt}$. The relations describe agents' uncertainty: $q \sim_a q'$ means that, while the system is in state q, agent a considers it possible that it is in q' now. Each $\sim_a$ is assumed to be an equivalence. It is required that agents have the same choices in indistinguishable states: if $q \sim_a q'$ then $d(a,q) = d(a,q')$.

Again, a *strategy* of an agent a is a conditional plan that specifies what a is going to do in each possible state. An executable (deterministic) plan must prescribe the same choices for indistinguishable states. Therefore ATL$_{ir}$ restricts the strategies that can be used by agents to the set of so called uniform strategies. A *uniform strategy* of an agent a is defined as a function $s_a : St \to Act$, such that: (1) $s_a(q) \in d(a,q)$, and (2) if $q \sim_a q'$ then $s_a(q) = s_a(q')$. A *collective strategy* for a group of agents $A = \{a_1, ..., a_r\}$ is a tuple of strategies $S_A = \langle s_{a_1}, ..., s_{a_r} \rangle$, one per each agent from A. A collective strategy is uniform if it contains only uniform individual strategies. Again, function $out(q, S_A)$ returns the set of all paths that may result from agents A executing strategy S_A from state q onward. The semantics of cooperation modalities in ATL$_{ir}$ is defined as follows:

$M, q \models \langle\langle A \rangle\rangle_{ir} \bigcirc \varphi$ iff there is a uniform collective strategy S_A such that, for each $a \in A$, q' such that $q \sim_a q'$, and path $\lambda \in out(S_A, q')$, we have $M, \lambda[1] \models \varphi$;

$M, q \models \langle\langle A \rangle\rangle_{ir} \Box \varphi$ iff there exists a uniform S_A such that, for each $a \in A$, q' such that $q \sim_a q'$, and $\lambda \in out(S_A, q')$, we have $M, \lambda[i] \models \varphi$ for each $i \geq 0$;

$M, q \models \langle\langle A \rangle\rangle_{ir} \varphi \mathcal{U} \psi$ iff there exist a uniform strategy S_A such that, for each $a \in A$, q' such that $q \sim_a q'$, and $\lambda \in out(S_A, q')$, there is $i \geq 0$ for which $M, \lambda[i] \models \psi$, and $M, \lambda[j] \models \varphi$ for each $0 \leq j < i$.

That is, $\langle\langle A \rangle\rangle_{ir} \varphi$ if agents A have a uniform strategy, such that for each path *that can possibly result from execution of the strategy according to at least one agent from* A, φ is the case.

Example 35 (Gambling robots) *Two robots (a and b) play a simple card game. The deck consists of Ace, King and Queen (A, K, Q). Normally, it is assumed that A is the best card, K the second best, and Q the worst. Therefore A beats K and Q, K beats Q, and Q beats no card. At the beginning of the game, the "environment" agent deals a random card to both robots (face down), so that each player can see his own hand, but he does not know the card of the other player. Then robot a can exchange his card for the one remaining in the deck (action $exch$), or he can keep the current one ($keep$). At the same time, robot b can change the priorities of the cards, so that Q becomes better than A (action chg) or he can do nothing (nop), i.e. leave the priorities unchanged. If a has a better card than b after that, then a win is scored, otherwise the game ends in a "losing" state. A CGS M_1 for the game is shown in Figure 6.3.*

It is easy to see that $M_1, q_0 \models \neg\langle\!\langle a \rangle\!\rangle_{ir} \Diamond \mathsf{win}$, because, for each a's (uniform) strategy, if it guarantees a win in e.g. state q_{AK} then it fails in q_{AQ} (and similarly for other pairs of indistinguishable states). Let us also observe that $M_1, q_0 \models \neg\langle\!\langle a, b \rangle\!\rangle_{ir} \Diamond \mathsf{win}$: in order to win, a must exchange his card in state q_{QK}, so he must exchange his card in q_{QA} too (by uniformity), and playing $exch$ in q_{QA} leads to the losing state. On the other hand, $M_1, q_{AQ} \models \langle\!\langle a, b \rangle\!\rangle_{ir} \bigcirc \mathsf{win}$ (a winning strategy: $s_a(q_{AK}) = s_a(q_{AQ}) = s_a(q_{KQ}) = keep$, $s_b(q_{AQ}) = s_b(q_{KQ}) = s_b(q_{AK}) = nop$; q_{AK}, q_{AQ}, q_{KQ} are the states that must be considered by a and b in q_{AQ}). Still, $M_1, q_{AK} \models \neg\langle\!\langle a, b \rangle\!\rangle_{ir} \bigcirc \mathsf{win}$.

Schobbens [121] proved that ATL$_{ir}$ model checking is **NP**-hard and $\mathbf{\Delta_2^P}$-easy. He also conjectured that the problem might be $\mathbf{\Delta_2^P}$-complete. We discuss the issue in more detail, and formally confirm his intuition in Section 6.6.

Remark 28 *The CTL universal path quantifier* A *can be expressed in ATL$_{ir}$ in the following way: $\mathsf{A}\varphi \equiv \langle\!\langle \varnothing \rangle\!\rangle_{ir} \varphi$. The existential path quantifier* E*, however, is not fully expressible when cooperation modalities quantify over uniform strategies only. Like for non-deterministic models, it may be the case that there is a single path for which property φ holds (i.e., we have $\mathsf{E}\varphi$), and yet even the "grand coalition" of agents $\mathbb{A}\mathrm{gt}$ is not able to enforce it (because $\mathbb{A}\mathrm{gt}$ can now use only uniform strategies), so $\langle\!\langle \mathbb{A}\mathrm{gt} \rangle\!\rangle_{ir} \varphi$ does not hold. Moreover, $\mathsf{E}\varphi\,\mathcal{U}\,\psi$ cannot be expressed as a combination of $\mathsf{A}\varphi\,\mathcal{U}\,\psi$, $\mathsf{E}\Diamond\varphi$, $\mathsf{E}\Box\varphi$, $\mathsf{A}\Box\varphi$, $\mathsf{E}\bigcirc\varphi$, and $\mathsf{A}\bigcirc\varphi$ (cf. [94], and the remark at the end of Section 6.2.1).*

6.3 Complexity of ATL Model Checking Revisited

The model checking problem asks, given model M, state q in M, and formula φ, whether φ holds in M, q. Model checking of temporal logics is usually computationally cheaper than satisfiability checking or theorem proving, while often being at least as useful because the designer or user of a system can come up with a precise model of the system behaviour (e.g. a graph with all the actions that may be executed) in many cases. For ATL, model checking has been proved *linear in the size of the models and formulae*. This seems to be a very good property, but unfortunately it guarantees less than one could expect.

function $mcheck_1(M, \varphi)$**.**

Returns the set of states in model $M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, o \rangle$ for which formula φ holds.

case $\varphi \in \Pi$ **:** return $\pi(p)$
case $\varphi = \neg\psi$ **:** return $St \setminus mcheck_1(M, \psi)$
case $\varphi = \psi_1 \vee \psi_2$ **:** return $mcheck_1(M, \psi_1) \cup mcheck_1(M, \psi_2)$
case $\varphi = \langle\langle A \rangle\rangle \bigcirc \psi$ **:** return $pre_1(M, A, mcheck_1(M, \psi))$
case $\varphi = \langle\langle A \rangle\rangle \Box \psi$ **:**
  $Q_1 := St$; $Q_2 := mcheck_1(M, \psi)$; $Q_3 := Q_2$;
  while $Q_1 \not\subseteq Q_2$
  do $Q_1 := Q_2$; $Q_2 := pre_1(M, A, Q_1) \cap Q_3$ **od**;
  return Q_1
case $\varphi = \langle\langle A \rangle\rangle \psi_1 \mathcal{U} \psi_2$ **:**
  $Q_1 := \varnothing$; $Q_2 := mcheck_1(M, \psi_1)$;
  $Q_3 := mcheck_1(M, \psi_2)$;
  while $Q_3 \not\subseteq Q_1$
  do $Q_1 := Q_1 \cup Q_3$; $Q_3 := pre_1(M, A, Q_1) \cap Q_2$ **od**;
  return Q_1
end case

function $pre_1(M, A, Q)$**.**

Auxiliary function; returns the exact set of states Q' such that, when the system is in a state $q \in Q'$, agents A can cooperate and enforce the next state to be in Q.

return $\{q \mid \exists \alpha_A \forall \alpha_{\mathbb{A}\mathrm{gt} \setminus A}\, o(q, \alpha_A, \alpha_{\mathbb{A}\mathrm{gt} \setminus A}) \in Q\}$

Figure 6.4: The ATL model checking algorithm from [8]

6.3.1 Model Checking ATL: Easy or Hard?

It has been known for a long time that formulae of CTL can be checked in time linear with respect to the size of the model and the length of the formula [30]. One of the main results concerning ATL states that its formulae can also be model-checked in deterministic linear time.

Proposition 69 ([7, 8]) *The ATL model checking problem is PTIME-complete, and can be done in time* $\mathbf{O}(ml)$*, where* m *is the number of transitions in the model and* l *is the length of the formula.*

The ATL model checking algorithm from [7, 8] is presented in Figure 6.4.

While the result is certainly attractive, it should be kept in mind that it is only relative to the size of models and formulae, and these can be very large for most application domains. Indeed, it is well known that the number of states in a model is usually exponential in the size of a higher-level description of the problem domain for both CTL and ATL models. Consider, for example, a system whose state space is defined by r Boolean variables (binary attributes). Obviously, the number of global states in the system is $n = 2^r$. A more general approach is presented in [92], where the "higher level description" is defined in terms of so called *concurrent programs*, that

can be used for simulating Boolean variables, but also processes or agents acting in parallel. Each concurrent program $C = \langle C_1, ..., C_k \rangle$ implicitly generates a system of global states which is defined as the product automaton of C. The main result concerning model checking is that checking CTL formulae is **PSPACE**-complete in the size of the concurrent program (and the length of the formula) [92].[5]

Thus, there are basically two kinds of results regarding model checking CTL and ATL. On the one hand, the problem is computationally easy with respect to CTL/ATL models one uses when defining semantics (sometimes called *global state graphs* [30] or *explicit models* [102]). On the other hand, the problem is very hard with respect to more compact representations (e.g. concurrent programs), mainly because these representations unravel into exponentially large explicit models. As a concurrent program may be seen as a system involving k agents, this already shows that having multiple agents can make models (and model checking) explode *with respect to a high level description*. What we point out in this article is that the complexity of $\mathbf{O}(ml)$ includes potential intractability *even on the level of explicit models* if the size of models is defined in terms of states rather than transitions, and the number of agents is a parameter of the problem rather than a fixed value. We state the observation formally as follows.

Remark 29 *Let n be the number of states in an explicit ATL model M. It was already observed in [8] that the number of transitions in M is* not *bounded by n^2, because transitions are labeled with tuples of agents' choices. Here, we make the observation more precise.*

Let k denote the number of agents, and d the maximal number of available decisions per agent per state. Then, $m = O(nd^k)$. In consequence, the ATL model checking algorithms from [7, 8] run in time $O(nd^k l)$, and hence their complexity is exponential if the number of agents is a parameter of the problem.

Example 33 is quite illustrative in this respect. The state space refers to valuations of only three attributes (two binary, and one ternary), which yields 12 states. And the number of transitions is already 216, despite the fact that the system includes only three agents, and each agent has only two or three actions available at each state.

Remark 30 *Note that, for turn-based models, only one agent is playing at a time, so the number of transitions is $O(nd)$, and hence model checking can be done in time $O(ndl)$.*

Throughout the rest of Section 6.3, we report results on the complexity of model checking ATL formulae over concurrent game structures, with n, k, d, l as input parameters. We show that the problem is $\mathbf{\Sigma_2^P}$-complete for "Positive ATL", and we cite [95], where model checking of full ATL is proved to be $\mathbf{\Delta_3^P}$-complete. The results seem natural as soon as we re-formulate $M, q \models \langle\langle a_1, ..., a_r \rangle\rangle \bigcirc \varphi$ as $\exists(\alpha_1, ..., \alpha_r) \forall(\alpha_{r+1}, ..., \alpha_k)\ M, o(q, \alpha_1, ..., \alpha_k) \models \varphi$ which bears close resemblance to the problem of QSAT$_2$. Before we discuss them formally, however, we must make one more important remark.

[5] We also note in passing that, for *some* high-level system descriptions, even the computation of $\langle\langle A \rangle\rangle \bigcirc$ may require **PSPACE**, or even **NEXPTIME** [34, 35], but these results are not relevant for our discussion here.

Remark 31 *Note that the transition function o must be kept external to the model checking algorithm, or represented in a somehow "compressed" way. Otherwise the algorithm requires exponential amount of memory to store the function, and in consequence the problem is not even in PSPACE.*

One way to achieve this is to assume that the transition function is implemented as an external procedure (more precisely: deterministic Turing machine) that, given state q and actions $\alpha_1, ..., \alpha_k$, returns the value of $o(q, \alpha_1, ..., \alpha_k)$ in polynomial time.

Another way is to represent transitions in a more compact way. Laroussinie and colleagues (after an idea that we developed in [78]) propose the notion of an implicit concurrent game structure. *An implicit CGS [95] is a CGS where, in each state q, the transition is defined by a finite sequence $((\varphi_1, q_1), ..., (\varphi_n, q_n))$. In the sequence, each q_i is a state, and each φ_i is a boolean combination of propositions $\hat{\alpha}^a$, where $\alpha \in d(a, q)$; $\hat{\alpha}^a$ stands for "agent a chooses action α". The transition function is now defined as follows:*

$$o(q, \alpha_1, ..., \alpha_k) = q_i \text{ iff } i \text{ is the lowest index such that } \{\hat{\alpha_1}^1, ..., \hat{\alpha_k}^k\} \models \varphi_i.$$

It is required that $\varphi_n \equiv \top$, so that no agent can enforce a deadlock. Every CGS can be encoded as an implicit CGS, with each φ_i being of polynomial size with respect to the number of states and actions [95].

6.3.2 "Positive ATL" Model Checking for CGS is Σ_2^{P}-hard

Firstly, we show that model checking of "positive" ATL formulae over concurrent game structures is $\mathbf{\Sigma_2^P}$-hard. We show this through a polynomial reduction of QSAT$_2$ to the model checking problem.

Definition 45 (QSAT$_i$) *The satisfiability of quantified Boolean formulae with i alternations of quantifiers is defined as follows.*

Input: *k propositional variables $p_1, ..., p_k$ (partitioned into i sets $P_1, ..., P_i$), and a Boolean formula θ that includes no other variables.*

Problem: *QSAT$_i$ asks if $\exists P_1 \forall P_2 \exists P_3 \ldots \Delta P_i\ \theta$ (where $\Delta = \forall$ if i is even, and $\exists$ if i is odd), i.e. whether there is a valuation of propositions in P_1 such that, for all valuations of propositions in P_2, there exists a valuation of propositions in P_3 etc., such that θ is satisfied.*

We use θ as a symbol for the Boolean formula that appears in QSAT$_i$, to distinguish it from the ATL formula in the model checking problem which we usually denote by φ. QSAT$_i$ is known to be $\mathbf{\Sigma_i^P}$-complete [112]. Obviously, in QSAT$_2$, we have only two sets of variables: P_1 and P_2.

To obtain the reduction, we construct a concurrent game structure M with 3 states: $St = \{q_0, q_\top, q_\bot\}$, and k agents: $\mathbb{A}\mathrm{gt} = \{a_1, ..., a_k\}$ that "decide" at q_0 upon valuations of propositions $p_1, ..., p_k \in P_1 \cup P_2$, respectively. Thus, agent a_i can "declare" proposition p_i true (action $\top$) or false (action $\bot$). Every tuple of actions from $\mathbb{A}\mathrm{gt}$ corresponds to a valuation $v_1, ..., v_k$ of the propositions, and vice versa. Now, the transitions from q_0 are defined in the following way:

$$o(q_0, v_1, ..., v_k) = \begin{cases} q_\top & \text{if } v_1, ..., v_k \models \theta \\ q_\bot & \text{else} \end{cases}$$

Transitions from $q_\top$ and $q_\bot$ do not matter. Note that $v_1, ..., v_k \models \theta$ can be verified in time and space linear in $|\theta|$, so o has a polynomial representation with respect to the size of the original problem.[6] Finally, we define proposition sat to hold only in state $q_\top$. Note that the agents "controlling" propositions from P_1 can enforce the next state to be $q_\top$ if, and only if, they can declare such a valuation of "their" propositions that θ is satisfied regardless of the opponents' choices:

Lemma 15 *Let A be the group of agents "responsible" for propositions P_1, i.e. $a_i \in A$ iff $p_i \in P_1$. Then, $\exists P_1 \forall P_2\, \theta$ iff $M, q_0 \models \langle\langle A \rangle\rangle \bigcirc \mathsf{sat}$.*

Proposition 70 *Model checking formulae of "Positive ATL" over concurrent game structures is $\mathbf{\Sigma_2^P}$-hard.*

6.3.3 "Positive ATL" Model Checking for CGS is in $\mathbf{\Sigma_2^P}$

In order to demonstrate membership in $\mathbf{\Sigma_2^P}$ of the model checking problem, we show an algorithm that computes the set of states in which formula φ holds, and lies in $\mathbf{NP^{NP}}$. A careful analysis of the algorithms proposed in [7, 8] reveals that the intractability is due to the pre-image operator pre, which is called at most n times for each subformula of φ.[7] Indeed, as we saw in the previous section, checking what a coalition can enforce in a single step (e.g., $M, q \models \langle\langle A \rangle\rangle \bigcirc \mathsf{sat}$) lies very close to the standard $\mathbf{\Sigma_2^P}$-complete problem of QSAT$_2$.

We show that checking a more sophisticated "positive" formula of ATL is no more complex than this. The main idea of the algorithm is as follows.

1. We guess nondeterministically *all* the choices that will be needed for any call to function Pre (that is, for each coalition A that occurs in φ, and for each state $q \in St$). The choices are then stored in the array $choice$.

2. We employ the standard model checking algorithm from Figure 6.4 with one important modification: every time function $pre_2(M, A, Q_1)$ is called, it assumes the subsequent A's choices from the array, and checks whether $q \in pre_2(M, A, Q_1)$ by calling an **NP** oracle (*is there a response from the opposition in q that leads to a state outside Q_1?*) and inverting its answer.

The detailed algorithm is shown in Figures 6.5 and 6.6. As the number of iterations, as well as the number of calls to pre, in the algorithm from Figure 6.4 is $O(nl)$, we get a nondeterministic polynomial algorithm that makes calls to an **NP** oracle.

Lemma 16 *Function $mcheck_2$ defines a nondeterministic Turing machine that runs in time $O(nkl)$, making calls to an* **NP** *oracle. The size of the witness is $O(nkl)$. The oracle is a nondeterministic Turing machine that runs in time $O(n + k)$.*

[6] Note that, in fact, this is a simple example of an implicit CGS.
[7] We recall that n is the number of states in M.

function $mcheck_2(M, \varphi)$**;**
Returns the set of states in M, in which formula φ holds.

- assign cooperation modalities in φ with subsequent numbers $1, ..., c$;

  // note that $c \leq l$; by $c(\varphi)$, we denote the number of coop. modalities in φ

  // we will denote the coalition from the ith cooperation modality in φ as $\varphi[i]$

- for each $i = 1, ..., c$, assign the agents in $\varphi[i]$ with numbers $1, ..., k_c$;

  // note that $k_c \leq k$ and $k_c \leq l$

  // we will denote the jth agent in A with $A[j]$

- guess an array $choice$ such that, for each $i = 1, ..., c$, $q \in St$, and $j = 1, ..., k_c$, we have that $choice[i][q][j] \in d_{\varphi[i][j]}(q)$;

  // at this point, the optimal choices for all coalitions in φ are guessed

  // note that the size of $choice$ is $O(nkl)$

  // by $choice|_i$, we will denote the array $choice$ with rows $1, ..., i-1$ removed

- return $eval_2(M, \varphi, choice)$;

function $eval_2(M, \varphi, choice)$**;**
Returns the states in which φ holds, given choices for all the coalitions from φ.

case $\varphi \in \Pi$**:** return $\{q \mid \varphi \in \pi(q)\}$;
case $\varphi = \neg\psi$**:** return $Q \setminus eval_2(M, \psi, choice)$;
case $\varphi = \psi_1 \vee \psi_2$**:** return $eval_2(M, \psi_1, choice) \cup eval_2(M, \psi_2, choice|_{c(\psi_1)+1})$;
case $\varphi = \langle\langle A\rangle\rangle \bigcirc \psi$**:** return $pre_2(M, A, eval_2(M, \psi, choice|_2), choice[1])$;
case $\varphi = \langle\langle A\rangle\rangle \Box \psi$**:** $Q_1 := St$; $Q_2 := Q_3 := eval_2(M, \psi, choice|_2)$;
while $Q_1 \not\subseteq Q_2$ **do** $Q_1 := Q_1 \cap Q_2$; $Q_2 := pre_2(M, A, Q_1, choice[1]) \cap Q_3$
od;
return Q_1;
case $\varphi = \langle\langle A\rangle\rangle \psi_1 \mathcal{U} \psi_2$**:** $Q_1 := \varnothing$; $Q_2 := eval_2(M, \psi_1, choice|_2)$;
$Q_3 := eval_2(M, \psi_2, choice|_{c(\psi_1)+2})$;
while $Q_3 \not\subseteq Q_1$ **do** $Q_1 := Q_1 \cup Q_3$; $Q_3 := pre_2(M, A, Q_1, choice[1]) \cap Q_2$
od;
return Q_1;
end case

Figure 6.5: Nondeterministic algorithm for model checking formulae of "Positive ATL"; part I.

Proposition 71 *Model checking formulae of "Positive ATL" over concurrent game structures is in* $\mathbf{\Sigma_2^P}$.

The following theorem is an immediate corollary:

Theorem 29 *Model checking "Positive ATL" formulae over CGS is* $\mathbf{\Sigma_2^P}$*-complete with respect to the number of (global) states, actions and agents, and the length of*

function $pre_2(M, A, Q_1, thischoice)$**;**
Returns the set of states, for which the A's choices from $thischoice$ enforce that the next state is in Q_1, regardless of what agents from $\mathbb{A}\mathrm{gt} \setminus A$ do.

- $Q_2 := \varnothing$;
- for each $q \in St$: **if** $oracle_2(M, A, Q_1, thischoice, q) = yes$ **then** $Q_2 := Q_2 \cup \{q\}$ **fi**;
- return Q_2;

function $oracle_2(M, A, Q_1, thischoice, q)$**;**
Returns yes if, and only if, the A's choices from $thischoice$ in q enforce that the next state is in Q_1, regardless of what agents from $\mathbb{A}\mathrm{gt} \setminus A$ do.

- guess an array $resp$ such that, for each $a \in \mathbb{A}\mathrm{gt} \setminus A$, we have $resp[a] \in d_a(q)$;

  `// at this point, the most dangerous response from the opposition is guessed`

  `// note that the size of` $resp$ `is` $O(k)$

- **if** $o(q, thischoice[q], resp) \in Q_1$ **then** return yes **else** return no **fi**;

Figure 6.6: Nondeterministic algorithm for model checking formulae of "Positive ATL": part II.

the formula. It is $\mathbf{\Sigma_2^P}$*-complete even for formulae that include only one cooperation modality, and only the "nexttime" temporal operator* $\bigcirc$*.*

6.3.4 Full ATL

As pointed out by Laroussinie, Markey and Oreiby, the algorithm in Figures 6.5 and 6.6 can be easily adapted to handle arbitrary ATL formulae in time $\mathbf{\Delta_3^P}$. In that case, strategies are guessed for each cooperation modality separately (and it is not necessary to guess them in advance).

Proposition 72 *Model checking formulae of ATL over concurrent game structures is in* $\mathbf{\Delta_3^P}$*.*

Laroussinie et al. have also proved that the bound is tight.

Theorem 30 ([95]) *Model checking ATL over CGS is* $\mathbf{\Delta_3^P}$*-complete in the number of states, actions and agents in the model, and the length of the formula.*

6.4 Model Checking with Alternating Transition Systems

Model checking ATL over CGS is $\mathbf{\Delta_3^P}$-complete (and $\mathbf{\Sigma_2^P}$-complete for "positive" formulae) when the size of models is defined in terms of the number

of states, and the number of agents is a parameter of the problem. However, the transition function in a CGS refers to choices that are abstract, while in alternating transition systems the function already encodes some information about possible outcomes of actions. One obvious advantage is that, in an ATS, the transition function is already represented in a compact way: for n states, k agents and at most d decisions per agent and state, the size of function δ is $O(n^2kd)$, while an explicit transition table in a CGS may require $O(nd^k)$ memory cells in general. In this section, we show that using ATS results in some advantages in terms of model checking complexity: it still sits in the nondeterministic polynomial hierarchy, but one level lower. Firstly, we demonstrate that the model checking of "Positive ATL" is in **NP** in Section 6.4.1. Secondly, in Sections 6.4.2 and 6.4.3, we define a variant of the Boolean satisfiability problem that we call "single false clause SAT" (*Sfc-SAT*), prove that it is **NP**-complete, and present a reduction of *Sfc-SAT* to the model checking problem. In Section 6.4.4, we cite [95] again, where correct complexity results for full ATL were presented.

Modeling systems via ATS is usually troublesome in practice, mostly due to the "singleton" requirement. In Section 6.4.5 we point out that, if we relax the requirement and allow for nondeterministic ATS's, we obtain the same model checking complexity for a strictly larger class of models.

6.4.1 Model Checking "Positive ATL" over ATS is in NP

Unlike in concurrent game structures, choices in alternating transition systems already contain some information about which states can possibly be achieved through them. More precisely, α includes *all* the states that can be achieved through α. Had it contained *only* such states, checking if it enforces φ would have been easy (it would have been sufficient to check whether φ holds in all $q' \in \alpha$). However, the latter condition is not true in general. [51] introduces the notion of a *tight* ATS: all states q' to which no transition exists from q are removed from agents' choices at q (i.e. from the elements of $\delta(q,a)$ for all $a \in \mathbb{A}\mathrm{gt}$). Still, this is not enough for our purposes, because $\alpha \in \delta(q,a)$ may include states that are reachable from q in general, but not by executing α. In this section, we propose a stronger notion of tightness, and show a nondeterministic algorithm to model check "Positive ATL" formulae over such ATS's. We also present a nondeterministic algorithm to "tighten" an ATS, and point out how these algorithms can be combined to obtain a procedure that model checks "Positive ATL" formulae over arbitrary ATS's in nondeterministic polynomial time. In the following, we assume without loss of generality that $A = \{a_1, ..., a_r\}$ for some $r \leq k$.

Definition 46 *Let $\alpha_A = \langle \alpha_1, ..., \alpha_r \rangle$ be a collective choice of A at q, i.e. $\alpha_i \in \delta(q, a_i)$. State q' is α_A-reachable from q if there is a combination of responses from the rest of agents: $\alpha_{r+1}, ..., \alpha_k$, $\alpha_i \in \delta(q, a_i)$ such that $q' \in \alpha_1 \cap ... \cap \alpha_k$.*

Definition 47 *ATS M is* strongly tight *if, for each $q \in St, a \in \mathbb{A}\mathrm{gt}$, we have that for each $q' \in \alpha_a \in \delta(q,a)$, q' is α_a-reachable from q.*

Lemma 17 *Let M be strongly tight, $\alpha_1, ..., \alpha_r$ be choices of $a_1, ..., a_r$ at q, and $q' \in \alpha_1 \cap ... \cap \alpha_r$. Then q' is $\langle \alpha_1, ..., \alpha_r \rangle$-reachable from q.*

function $tighten(M)$**;**

For each $a_i \in \mathbb{A}\mathrm{gt}$, $q \in St$, $\alpha_i \in \delta(q, a_i)$, and $q' \in \alpha_i$:

- guess the "opposition" responses $\alpha_1, ..., \alpha_{i-1}, \alpha_{i+1}, ..., \alpha_k$;
- **if** $q' \notin \alpha_1 \cap ... \cap \alpha_k$ **then** remove q' from α_i;

Figure 6.7: Algorithm for "tightening" alternating transition systems

function $pre_3(M, A, Q1)$**;**

- $pre := \varnothing$;
- for each $q \in St$:
  - guess $\alpha_a \in \delta(q, a)$ for each $a \in A$;
  - **if** $\bigcap_{a \in A} \alpha_a \subseteq Q1$ **then** $pre := pre \cup \{q\}$;
- return pre;

Figure 6.8: New pre-image function for model checking ATL over alternating transition systems

Every ATS can be made strongly tight via the procedure in Figure 6.7. Moreover, "Positive ATL" formulae can be model-checked over strongly tight ATS's via the original ATL model checking algorithm from Figure 6.4, with function $pre(A, Q)$ implemented as in Figure 6.8. We observe that – if we assign numbers $1, ..., |\delta(q, a)|$ to choices from $\delta(q, a)$ for all q, a at the beginning, so that the choices are further identified by abstract labels rather than their content – all the "guessing" operations are independent from each other when we evaluate a "positive" formula. Thus, we can apply the same trick as in Section 6.3.3, and guess *all* the necessary information beforehand. The size of the witness is $O(n^2k^2d + nkl)$, hence we obtain an **NP** algorithm for the model checking.

Proposition 73 *Model checking formulae of "Positive ATL" over alternating transition systems is in* **NP**.

6.4.2 Single False Clause SAT

To prove **NP**-hardness of model checking "Positive ATL" over alternating transition systems, we define the following variant of the SAT problem.

Definition 48 (Sfc-SAT: single false clause SAT) Input: *(1)* n *clauses:* $C_1, ..., C_n$*, in* k *propositions:* $p_1, ..., p_k$ *such that for each valuation of* $p_1, ..., p_k$*,* exactly one clause is false*;*
(2) numbers $m \leq n$*,* $r \leq k$*.*

Problem: Is there a valuation of $p_1, ..., p_r$ such that all clauses $C_1|_r, ..., C_m|_r$

are satisfied? *Clause $C|_r$ is obtained from clause C by deleting all literals that refer to propositions $p_{r+1}, ..., p_k$ (we keep only the literals up to r).*

Remark 32 *Obviously,* Sfc-SAT *is in* **NP** *(it is sufficient to guess a valuation and check whether it is a good one).*

In order to show that *Sfc-SAT* is **NP**-hard, we show that 3-SAT can be reduced to it. In 3-SAT, we are given m clauses $C_1, ..., C_m$ over r propositions $p_1, ..., p_r$ such that each clause C_i contains at most three literals: $C_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$ ($l_{i,j}$ are p_l or $\neg p_l$, $1 \leq i \leq m$). This special instance of the satisfiability problem is also **NP**-complete [112]. Note that the m and the r are the respective numbers occurring as inputs in Definition 48. To show that 3-SAT can be reduced to *Sfc-SAT*, we demonstrate that there are propositions $p_{r+1}, \ldots, p_k$, and clauses $C'_1, ..., C'_n$, with $m \leq n$, $C_i \subseteq C'_i$ and $C'_i|_r = C_i$ for $i \leq m$, such that for each valuation of $p_1, \ldots, p_k$, exactly one of C'_i is false.

What does the last condition mean for a set of clauses $C'_1, ..., C'_n$? Basically, it means that these clauses represent all 2^k possibilities of choosing truth values for $p_1, \ldots, p_k$. So, the problem in the reduction is to *extend the given clauses by new variables* and to *add new clauses*. This has to be done so that the length of the new problem is still polynomial in the length of the given 3-SAT instance.

We assume without loss of generality that none of $C'_1, ..., C'_m$ contains a complementary pair of literals (otherwise the clause would be satisfiable under all valuations and could be safely discarded as it does not matter for the overall satisfiability problem). In order to extend clauses $C_1, ..., C_m$ in an appropriate way, we use auxiliary formulae α_i and β, defined in the following way:

α_i: We construct formulae α_i stating that *a selected clause number is* $i \leq m$. To be more precise, we introduce $t := \lceil \log m \rceil$ new variables $y_1, \ldots, y_t$ and define conjunctions α_i ($i = 1, \ldots, m$) over these variables as follows (this idea is due to Thomas Eiter [37]). We write each number $1, \ldots, m$ in binary and represent each (of the t) digits by the new variables (a 1 is represented by the variable itself, a 0 by the negation of the variable). The i'th digit is then represented by y_i if it is 1 and by $\neg y_i$ if it is 0. Thus, for each valuation of the new variables, only one conjunction α_i can be true, namely the one representing the number coded in the binary representation.

Note that we can also represent numbers greater than m (up to the next power of 2, namely 2^t). These conjunctions do not correspond to the m original clauses from the 3-SAT problem. In our reduction, we have to distinguish between them. Therefore we introduce a formula β in the next step.

β: We construct a formula β stating that the selected clause number is less than or equal to m. Thus, β satisfies the following equivalences: $\beta \Leftrightarrow \bigvee_{i=1}^{m} \alpha_i \Leftrightarrow \bigwedge_{i=m+1}^{2^{\lceil \log m \rceil}} \neg \alpha_i$.

We therefore define a set of clauses β, which describe all valuations

corresponding to numbers strictly greater than m. Thus we have:

$$\beta \Leftrightarrow \bigwedge_{i=1}^{m} \neg\alpha_i.$$

Realising β as a set of clauses is simple: we just take α_m and check that they coincide on an initial segment and then a negated variable occurs (where in α_m a positive variable is located).

β can also be written as a set of clauses

$$\beta \Leftrightarrow \bigwedge_{i=m+1}^{2^{\lceil \log m \rceil}} \neg\alpha_i.$$

Note that the last formula is a set clauses (because all $\neg\alpha_i$ are clauses), and hence we need at most $2^{\lceil \log m \rceil} - m$ many clauses to represent β (which is never more than m). We denote these clauses by $C_1^\beta, \ldots, C_m^\beta$. Each clause C_j^β states, that *the selected clause has not the number* $m + j$.

In the following we sometimes use β to represent the (at most) m clauses $C_1^\beta, \ldots, C_m^\beta$.

Extending the clauses: For each $C_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$ we construct the *remaining* 7 clauses (all parities of the 3 variables) and add $\neg\alpha_i$. So, for each C_i we get 8 clauses $C'_{i,0}, \ldots C'_{i,7}$, where $C'_{i,0} = C_i \vee \neg\alpha_i$ and $(C'_{i,0} \wedge \ldots \wedge C'_{i,7}) \Leftrightarrow \neg\alpha_i$. Note, again, that $\neg\alpha_i$ is always a clause. We observe also that the m clauses $C_1, ..., C_m$, which we originally started with (as an instance of 3-SAT), are, by construction, exactly $C'_{1,0}|_r, C'_{2,0}|_r, \ldots, C'_{m,0}|_r$.

Reduction: The (at most) $s := m + m \times 8$ clauses:

$$C_1^\beta, \ldots, C_m^\beta, \qquad \text{and} \qquad C'_{i,j} \ (1 \leq i \leq m, 0 \leq j \leq 7),$$

over $k = r + \lceil \log m \rceil$ variables, represent an instance of *Sfc-SAT*, such that if we choose $m \leq n$ and $r \leq k$, then we get the 3-SAT problem we started with.

Why are the clauses above an instance of *Sfc-SAT*? The fact that we get back the 3-SAT problem has already been shown. It is also obvious that the constructed instance is polynomial in the size of the instance we started with. So it remains to show that for each valuation of all the variables, *exactly one clause is false*. Let a valuation be given. We must consider two cases:

1. Exactly one of the $\alpha_1, \ldots, \alpha_m$ is true, say α_{i_0} (this is decided by the newly introduced variables). Then all clauses $C'_{i,j}$ with $i \neq i_0$ are true (because $\neg\alpha_i$ is true and it occurs as a disjunct in all these clauses). Of the 8 clauses $C'_{i_0,j}$ $(0 \leq j \leq 7)$, exactly one is false, namely the one contradicting the valuation of the three old variables occurring in the original C_i (note that all possibilities are covered with the 8 cases). Clearly, β (i.e all clauses C_j^β) is true as well.

2. None of the $\alpha_1, \ldots, \alpha_m$ is true. But then all clauses $C'_{i,j}$ are true and only β is false, i.e. exactly one of the clauses C_j^β.

These are all the cases, because α_i (resp. C_j^β) are pairwise inconsistent by construction: any two different conjunctions α_i, α_j (resp. C_i^β, C_j^β) with $i \neq j$ contain at least one pair of complementary literals. This gives us the following result:

Proposition 74 *Sfc-SAT is* **NP***-complete.*

6.4.3 Reduction of Sfc-SAT to "Positive ATL" Model Checking over ATS

To obtain the reduction, we construct an ATS M with $St = \{q_0, C_1, ..., C_n\}$, i.e. one state per clause plus an initial state. Next, we "simulate" propositions $p_1, ..., p_k$ with agents $a_1, ..., a_k$. Each agent "declares" his proposition true or false in the initial state q_0. Thus, agent a_i has two available choices at q_0: to declare p_i true or to declare p_i false; a choice of a_i is represented with the set of clauses that are *not* made true by setting the value of p_i in this particular way. For example, for clauses $C_1 = p_1 \vee \neg p_2, C_2 = p_2$, a_1's choices are represented as $\{C_2\}, \{C_1, C_2\}$: if p_1 is set to true, only C_2 can be false, but if p_1 is set to false, both C_1, C_2 can remain unsatisfied. Choices and transitions at states $C_1, ..., C_m$ do not really matter. There is only one atomic proposition, $\mathsf{therest}$, with $\pi(\mathsf{therest}) = \{C_{m+1}, ..., C_n\}$.

Note that each combination of choices from $a_1, ..., a_k$ at q_0 corresponds to a single valuation of $p_1, ..., p_k$, and vice versa. Moreover, a clause is not satisfied by a valuation iff no proposition "makes" it true. Thus, the set of clauses, unsatisfied by a valuation, is equal to the intersection of sets of clauses that are not "made" true by each single proposition. By definition of *Sfc-SAT*, such an intersection is always a singleton, which proves that M is indeed an ATS.

Lemma 18 *There is a valuation of $p_1, ..., p_r$ such that all clauses $C_1|r, ..., C_m|r$ are satisfied iff $M, q_0 \models \langle\langle a_1, ..., a_r\rangle\rangle \bigcirc \mathsf{therest}$.*

Proof **[⇒]** Suppose that there is such a valuation of $p_1, ..., p_r$. Thus, regardless of the actual valuation of $p_{r+1}, ..., p_k$, clauses $C_1, ..., C_t$ must be true, and hence for each valuation of $p_{r+1}, ..., p_k$, the single unsatisfied clause must be among $C_{t+1}, ..., C_n$. Re-writing it in terms of the ATS M: there is a collective choice of $a_1, ..., a_r$ such that, for each tuple of choices from the other agents, the resulting next state must be among $C_{t+1}, ..., C_n$. In consequence, $M, q_0 \models \langle\langle a_1, ..., a_r\rangle\rangle \bigcirc \mathsf{P}$.

[⇐] Let $M, q_0 \models \langle\langle a_1, ..., a_r\rangle\rangle \bigcirc \mathsf{P}$. Thus, there is a collective choice $S_{\{a_1,...,a_r\}}$ such that, for each tuple of choices from the other agents, the resulting next state is always among $C_{t+1}, ..., C_n$. We take the valuation of $p_1, ..., p_r$ that corresponds to this $S_{\{a_1,...,a_r\}}$. By the shape of the construction, each $C_i \in \{C_1, ..., C_t\}$ must be true for each valuation of $p_{r+1}, ..., p_k$. In particular, C_i is true for the valuation $p_{r+1} = \bot, ..., p_k = \bot$. Thus, $C_i|r$ is also true. ■

Note that the reduction can be done in time polynomial in n, k. Computing the agents' choice sets is the hardest point here, and it can be done in time $O(k^2 n)$. The resulting model includes $n + 1$ states, k agents, and

$d = 2$ choices per agent per state – and the length of the resulting formula is $l = r + 2 \leq k + 2$, which concludes the reduction, and proves that the model checking problem is **NP**-hard. Thus, we have the following.

Theorem 31 *Model checking "Positive ATL" formulae over ATS is* **NP***-complete with respect to the number of (global) states, actions and agents, and the length of the formula. It is* **NP***-complete even for formulae that include only one cooperation modality, and only the "nexttime" temporal operator* $\bigcirc$.

6.4.4 Full ATL

Like in the case of CGS, the algorithm presented in Section 6.4.1 can be easily adapted to model-check arbitrary ATL formulae in time $\mathbf{\Delta_2^P}$. Again, strategies are guessed for each cooperation modality separately (and it is not necessary to guess them in advance).

Proposition 75 *Model checking formulae of ATL over alternating transition systems is in* $\mathbf{\Delta_2^P}$.

Laroussinie, Markey and Oreiby have proved that the bound is tight.

Theorem 32 ([95]) *Model checking ATL over ATS is* $\mathbf{\Delta_2^P}$*-complete in the number of states, actions and agents in the model, and the length of the formula.*

6.4.5 Model Checking with Nondeterministic Transition Systems

Alternating transition systems were proposed as models for open computational systems, and the way in which the transition function is constructed reflects this intention. The problem with ATS's is that they are *not* modular, partly due to the "singleton intersection" requirement: legality of a choice cannot be defined in isolation from the rest of the choices in a given state. Adding another process to the system usually requires thorough reconstruction of the model: in particular, new states must be added, and agents' choices extended so that *each* intersection is again a singleton. We suggest that the requirement can be relaxed, yielding a more general (and more flexible) class of models with the same ATL model checking complexity. To show this, we define *non-deterministic alternating transition systems* (NATS) in the same way as ATS, except that no requirement on function δ is imposed.[8] Obviously, model checking ATL formulae over NATS is $\mathbf{\Delta_2^P}$-hard (and **NP**-hard for "positive" formulae), because ATS are special cases of NATS. Moreover, the model checking algorithm, depicted in Section 6.4.1, can be applied to NATS as well.

Theorem 33 *Model checking ATL formulae over NATS is* $\mathbf{\Delta_2^P}$*-complete for the full language, and* **NP***-complete for the "positive" sublanguage.*

[8] Traditionally, the transition relation is required to be serial in models of temporal logic, in order to make sure that the time "flows forever". This can be ensured by requiring that, for each tuple of choices $\alpha_i \in \delta(q, a_i)$, the intersection $\alpha_1 \cap ... \cap \alpha_k$ is non-empty. Our argument in this section is valid for such a variant of NATS, too.

This suggests that using the more general class of NATS may be beneficial for most purposes: we can get rid of the rigid and highly inconvenient "singleton" requirement without any computational cost! But, as we already pointed out in Sections 6.2.1 and 6.2.2, the existential path quantifier E from CTL cannot be fully embedded in ATL, when the latter has its semantics defined over NATS. This looks as a serious shortcoming in terms of expressivity at first glance. However, we observe that we can deal with this problem by adding another temporal operator to the language of ATL. The operator we propose to add is the "weak until" operator $\mathcal{W}$, known in temporal logic for a long time [99], although not as popular as the "strong until" $\mathcal{U}$. Formula $\varphi \mathcal{W} \psi$ means that, if ψ becomes eventually true, then φ holds until the first occurrence of ψ, otherwise φ holds for ever.[9] The formal semantics of "weak until" can be defined as follows:

$M, q \models \langle\langle A \rangle\rangle \varphi \mathcal{W} \psi$ iff there exists S_A such that, for each $\lambda \in out(S_A, q)$: (1) there is $i \geq 0$ for which $M, \lambda[i] \models \psi$, and $M, \lambda[j] \models \varphi$ for each $0 \leq j < i$, or (2) $M, \lambda[j] \models \varphi$ for each $j \geq 0$.

In Figure 6.9, we present a simple extension of the model checking algorithm from Figure 6.4 that deals with "weak until" formulae in a way analogous to model checking $\langle\langle A \rangle\rangle \Box \varphi$ and $\langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$. The model checking algorithm from Section 6.4.1 can be augmented in the same way. It is easy to see that the complexity of the algorithms stays the same as before. Moreover, we show that adding $\langle\langle A \rangle\rangle \varphi \mathcal{W} \psi$ to ATL allows to express the full power of CTL (and more), through the following translation:

$$
\begin{aligned}
\mathsf{A}\bigcirc \varphi &\equiv \langle\langle \varnothing \rangle\rangle \bigcirc \varphi \\
\mathsf{A}\varphi \mathcal{U} \psi &\equiv \langle\langle \varnothing \rangle\rangle \varphi \mathcal{U} \psi \\
\mathsf{A}\varphi \mathcal{W} \psi &\equiv \langle\langle \varnothing \rangle\rangle \varphi \mathcal{W} \psi \\
\mathsf{E}\bigcirc \varphi &\equiv \neg \mathsf{A} \bigcirc \neg \varphi \\
\mathsf{E}\varphi \mathcal{U} \psi &\equiv \neg \mathsf{A}(\neg \psi) \mathcal{W} (\neg \varphi \wedge \neg \psi) \\
\mathsf{E}\varphi \mathcal{W} \psi &\equiv \neg \mathsf{A}(\neg \psi) \mathcal{U} (\neg \varphi \wedge \neg \psi) \\
\Diamond \varphi &\equiv \top \mathcal{U} \varphi \\
\Box \varphi &\equiv \varphi \mathcal{W} \bot
\end{aligned}
$$

Note that formulae $\langle\langle A \rangle\rangle \Box \varphi$ do not have to be included in the definition of ATL explicitly any more, since the $\Box$ operator can be derived from $\mathcal{W}$.

Theorem 34 *ATL with "weak until" covers the full expressive power of CTL even when nondeterministic alternating transition systems are used as models. Moreover, model checking ATL with "weak until" over nondeterministic ATS is:*

1. **P***-complete (linear time) with respect to the number of transitions in the model and the length of the formula;*

[9] We note that "weak until" is not expressible even in ATL with deterministic transitions, cf. [95].

case $\varphi = \langle\langle A\rangle\rangle \psi_1 \mathcal{W} \psi_2$ **:**
$Q_1 := mcheck(M, \psi_1) \cup mcheck(M, \psi_2)$;
$Q_2 := St$;
$Q_3 := mcheck(M, \psi_2)$;
while $Q_2 \neq Q_1$
do $Q_2 := Q_1$; $\quad Q_1 := Q_2 \setminus \big((St \setminus pre(M, A, Q_2)) \setminus Q_3\big)$ **od**;
return Q_1

Figure 6.9: Subroutine for model checking "weak until"

2. **NP**-*complete for the "positive" sublanguage with respect to the number of states, agents and decisions in the model and the length of the formula;*

3. $\mathbf{\Delta_2^P}$-*complete for the full language with respect to the number of states, agents and decisions in the model and the length of the formula.*

Finally, we observe that ATS have already been used in the work on implementing symbolic model checking for ATL [87], probably because of the compact representation of the transition function.[10] We proved in this section that using ATS offers also some computational advantage over CGS. Theorem 34 suggests that *designing* ATS does not have to be such a painstaking process.

6.5 Turning Game Models Turn-Based

In this section, we demonstrate how strategic ability in arbitrary ATS's can be translated into strategic ability in turn-based systems. More precisely, we show how, for an arbitrary alternating transition system M, a turn-based system M' can be constructed, so that a combination of *choices* in M corresponds to a combination of *strategies* in a fragment of M'. We then propose a translation of ATL formulae into ATL$^+$ formulae, such that the original formula holds in M, q if, and only if, the translated formula holds in M', q. Finally, we point out that the latter can be model-checked in nondeterministic polynomial time ($\mathbf{\Delta_2^P}$ for full ATL, and **NP** for "Positive ATL"), and provide another (slightly more general) proof of the upper bounds for both variants of the language.

The translation of models is independent from the translation of formulae in our construction, which allows for "pre-compiling" models when one wants to check various properties of a particular multi-agent system.

[10] The authors of [87] define the semantics of ATL in terms of concurrent game structures, but the model checking algorithm they present uses *postconditions* to specify the possible outcomes of a choice. A postcondition is taken to be simply a set of states, and choices by different agents executed in parallel lead to the state from the intersection of the postconditions (it is even assumed that the intersection must be a singleton).

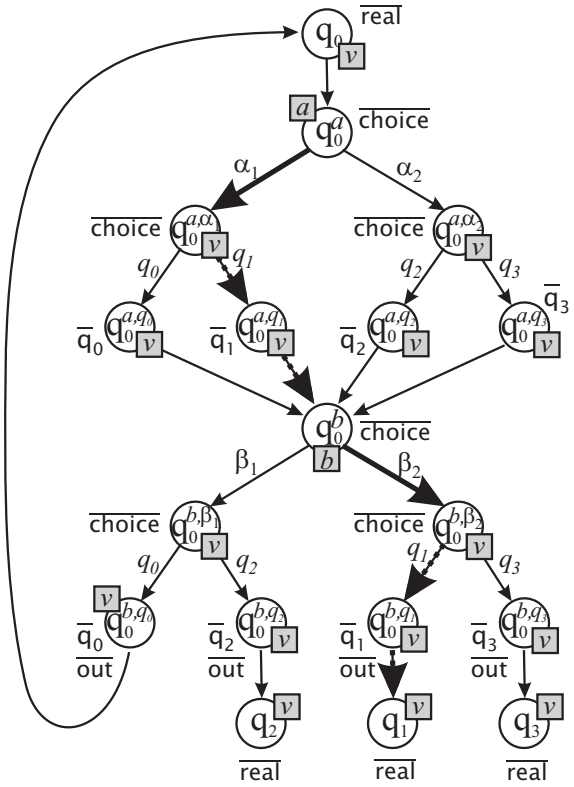


Figure 6.10: A fragment of M_1': simulation of outgoing transitions from q_0

6.5.1 Translation of Models

Let $M = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, \delta \rangle$ be an ATS. We construct a turn-based ATS $M' = \langle \mathbb{A}\mathrm{gt}', St', \Pi', \pi', \delta' \rangle$ as follows:

- $\mathbb{A}\mathrm{gt}' = \mathbb{A}\mathrm{gt} \cup \{\mathbf{v}\}$: we add an additional agent $\mathbf{v}$ ("verifier") to the original set of players. Verifier helps to find out the right outcome state, given the choices from all agents (i.e. the sole state which belongs to the intersection of their choices);
- $St' = St \cup \bigcup_{a \in \mathbb{A}\mathrm{gt}} (dec(a) \cup exec(a) \cup outcome(a))$, where:
  - $dec(a) = \{q^a \mid q \in St\}$ are the "dummy states" from which agent a's decisions are simulated; by x^ρ, we will denote a copy of item x, labeled with superscript ρ.
  - $exec(a) = \{q^{a,S} \mid q \in St, S \in \delta(q,a)\}$ simulate the situations between a's decision making and the execution of a decision.
  - $outcome(a) = \{q^{a,q'} \mid q \in St, q' \in \bigcup_{S \in \delta(q,a)} S\}$ are the dummy states that simulate possible outcomes of a's decisions.
- $\Pi' = \{\overline{\mathsf{q}} \mid q \in St\} \cup \{\overline{\mathsf{real}}, \overline{\mathsf{choice}}, \overline{\mathsf{out}}\}$. Proposition $\overline{\mathsf{real}}$ marks the original, "real" states from M; $\overline{\mathsf{choice}}$ labels the dummy states that simulate situations before and after a choice, $\overline{\mathsf{out}}$ marks the *final* outcome states

before the next "real" state is reached, and $\overline{\mathsf{q}_\mathsf{i}}$ mark "outcome" dummy states that refer to a transition ending up in state q_i. Thus:

- $\pi'(\overline{\mathsf{real}}) = St$, $\pi'(\overline{\mathsf{choice}}) = \bigcup_{a\in\mathbb{A}\mathrm{gt}}(dec(a) \cup exec(a))$,
- $\pi'(\overline{\mathsf{out}}) = outcome(a_k)$,
- $\pi'(\overline{\mathsf{q}_\mathsf{i}}) = \{q^{a,q_i} \mid q^{a,q_i} \in outcome(a), a \in \mathbb{A}\mathrm{gt}\}$.

- The "decision" states are "owned" by the decision making players; the rest of the states is owned by verifier:
  - $\delta(q, \mathbf{v}) = \{St'\}$ for $q \in dec(a), a \in \mathbb{A}\mathrm{gt}$,
  - $\delta(q, a) = \{St'\}$ for $q \notin dec(a)$.
- Choices of the original agents remain the same as in M, but they are split between "choice" states. Verifier makes substantial choice only at the "execution" dummy states. Transitions from the "outcome" dummy states are automatic, and lead to the decision node of the next player. Choices executed by agents at decision nodes lead to their corresponding execution states, and verifier's actions at execution nodes lead to their corresponding outcome nodes.
  - $\delta(q^a, a) = \{\{q^{a,Q_1}\}, ..., \{q^{a,Q_i}\}\}$ for $q^a \in dec(a)$, $\delta(q, a) = \{Q_1, ..., Q_i\}$;
  - $\delta(q^{a,S}, \mathbf{v}) = \{\{q^{a,q_1}\}, ..., \{q^{a,q_i}\}\}$ for $q^{a,S} \in exec(a)$ and $S = \{q_1, ..., q_i\}$.
  - $\delta(q^{a_i,q_j}, \mathbf{v}) = \{\{q^{a_{i+1}}\}\}$ for $i < k$, and $\delta(q^{a_k,q_j}, \mathbf{v}) = \{\{q^{a_k}\}\}$.

Example 36 *Consider a fragment of the alternating transition system M_2, depicted in Figure 6.2. The fragment of the resulting ATS M_2', that refers to the transitions starting from q_0, is shown in Figure 6.10. (Remember, we use symbols α_1, α_2 and β_1, β_2 as shorthand for the choices to make the example easier to read, but in fact these are sets of states and not abstract labels.) The collective strategy of $\{a, b\}$, that corresponds to the combination of choices $\langle\alpha_1, \beta_2\rangle$ in the original ATS, is marked with bold arrows. The only verifier's response, that yields a path with exactly one $\overline{\mathsf{q}_\mathsf{i}}$ proposition holding along it, is also indicated.*

Note that, for each state q in M, the transformation of the outgoing transitions requires that we process all the choices from $\delta(q, a)$ once; we must also process the "contents" of each choice (i.e. all the states included in the choice) – but only once, too. Moreover, the resulting substructure includes at most $O(kdn)$ outgoing transitions per node.

Proposition 76 *The translation of M can be done in time $O(n^2kd)$, and M' includes $m' = O(n^2kd)$ states.*

6.5.2 Translation of Formulae

Let φ, ψ be ATL formulae, whose interpretations in M are $[\![\varphi]\!], [\![\psi]\!]$ respectively.[11] We define the translation of complex formulae in the following way:

[11] We will abuse the notation slightly by using $[\![\varphi]\!]$ to denote also $\bigvee_{q_i\in[\![\varphi]\!]} \overline{\mathsf{q}_\mathsf{i}}$, a formula that holds exactly in the states from $[\![\varphi]\!]$.

$$
\begin{array}{rcl}
tr_M(\neg\varphi) & = & \neg[\![\varphi]\!] \\
tr_M(\varphi \wedge \psi) & = & [\![\varphi]\!] \wedge [\![\psi]\!] \\
next_M(\varphi) & = & \neg \bigvee_{q_i \notin [\![\varphi]\!]} (\overline{\mathsf{q}_\mathsf{i}} \vee \overline{\mathsf{real}} \vee \overline{\mathsf{choice}})\,\mathcal{U}\ \overline{\mathsf{out}} \\
tr_M(\langle\!\langle A\rangle\!\rangle \bigcirc \varphi) & = & \langle\!\langle A\rangle\!\rangle next_M(\varphi) \\
tr_M(\langle\!\langle A\rangle\!\rangle \Box \varphi) & = & [\![\varphi]\!] \wedge \langle\!\langle A\rangle\!\rangle \Box(\overline{\mathsf{real}} \rightarrow \langle\!\langle \varnothing\rangle\!\rangle next_M(\varphi)) \\
tr_M(\langle\!\langle A\rangle\!\rangle \varphi\,\mathcal{U}\,\psi) & = & [\![\psi]\!] \vee ([\![\varphi]\!] \wedge \langle\!\langle A\rangle\!\rangle(\overline{\mathsf{real}} \rightarrow \langle\!\langle \varnothing\rangle\!\rangle next_M(\varphi))\,\mathcal{U}\ [\![\psi]\!]).
\end{array}
$$

The idea is as follows: the paths that matter are the ones where only a single proposition $\overline{\mathsf{q}_\mathsf{i}}$ occurs in each subpath between two subsequent "real" states – they correspond to intersections of the agents' choices that can be found along the subpath. For $\langle\!\langle A\rangle\!\rangle \bigcirc \varphi$, we want to make sure that A have a strategy to enforce that all such subpaths until the next "real" node refer to states from $[\![\varphi]\!]$. In other words, A must have a strategy such that *no* initial subpath occurs that refers to some $q_i \notin [\![\varphi]\!]$. For $\langle\!\langle A\rangle\!\rangle \Box \varphi$, the same must hold for subpaths after *each* "real" node etc. Note that $tr_M(\Phi)$ is a formula of ATL$^+$, since it includes Boolean combinations of temporal formulae.[12] The following proposition states that the translation is correct.

Proposition 77 *Let φ be an ATL formula that does not include special propositions $\overline{\mathsf{real}}$, $\overline{\mathsf{out}}$, $\overline{\mathsf{choice}}$ and $\overline{\mathsf{q}_\mathsf{i}}$. Let M be an ATS, and q a state in M. Then:*

$$M, q \models \Phi \quad \textit{iff} \quad M', q \models tr_M(\Phi).$$

Proof We prove the proposition for the case $\Phi \equiv \langle\!\langle A\rangle\!\rangle \bigcirc \varphi$. The other cases follow from respective fixpoint characterisations of $\langle\!\langle A\rangle\!\rangle \Box \varphi$ and $\langle\!\langle A\rangle\!\rangle \varphi\,\mathcal{U}\,\psi$.

[⇒] Let $M, q \models \langle\!\langle A\rangle\!\rangle \bigcirc \varphi$, $A = \{a_1, ..., a_r\}$. Suppose that $M', q \not\models \langle\!\langle A\rangle\!\rangle \neg \bigvee_{q_i \notin [\![\varphi]\!]} (\overline{\mathsf{q}_\mathsf{i}} \vee \overline{\mathsf{real}} \vee \overline{\mathsf{choice}})\,\mathcal{U}\ \overline{\mathsf{out}}$. Note that $\overline{\mathsf{out}}$ holds for the first time exactly after $3k$ transitions from q in M'. Thus, for each strategy S'_A in M' there is a path $\lambda \in out(q, S'_A)$, and a state $q_i \notin [\![\varphi]\!]$, such that $M, \lambda[j] \models (\overline{\mathsf{q}_\mathsf{i}} \vee \overline{\mathsf{real}} \vee \overline{\mathsf{choice}})$ for all $j = 0, ..., 3k-1$. We take any strategy S_A in M, find the corresponding S'_A with $s'_a(q^a) = s_a(q)$ for $a \in A$, and then we take the above λ and q_i. We set the choices of the opponents in M, q to $s_{a_j}(q) = \sigma$ such that $\lambda[3j-2] = q^{a_j,\sigma}$, $a_j \notin A$. By construction, $q_i \in s_{a_1}(q) \cap ... \cap s_{a_k}(q)$, which gives a contradiction.

[⇐] Similarly: we take the "winning" strategy in M', construct the corresponding strategy in M (or rather its relevant part for state q), and show that no combination of responses from $a_{r+1}, ..., a_k$ can lead to a state $q' \notin [\![\varphi]\!]$. ■

Example 37 *Consider models M_2 and M'_2 again. Formula $\langle\!\langle a\rangle\!\rangle \bigcirc (\mathsf{p}_1 \vee \mathsf{p}_2)$ holds in M_2, q_0, and indeed $M'_2, q^a_0 \models \langle\!\langle a\rangle\!\rangle \neg\big((\overline{\mathsf{q}_2} \vee \overline{\mathsf{real}} \vee \overline{\mathsf{choice}})\,\mathcal{U}\ \overline{\mathsf{out}} \vee (\overline{\mathsf{q}_3} \vee \overline{\mathsf{real}} \vee \overline{\mathsf{choice}})\,\mathcal{U}\ \overline{\mathsf{out}}\big)$. On the other hand, $M'_2, q_0 \not\models \langle\!\langle a\rangle\!\rangle \bigcirc \mathsf{p}_1$, and $M'_2, q^a_0 \not\models \langle\!\langle a\rangle\!\rangle \neg\big((\overline{\mathsf{q}_1} \vee \overline{\mathsf{real}} \vee \overline{\mathsf{choice}})\,\mathcal{U}\ \overline{\mathsf{out}} \vee (\overline{\mathsf{q}_2} \vee \overline{\mathsf{real}} \vee \overline{\mathsf{choice}})\,\mathcal{U}\ \overline{\mathsf{out}} \vee (\overline{\mathsf{q}_3} \vee \overline{\mathsf{real}} \vee \overline{\mathsf{choice}})\,\mathcal{U}\ \overline{\mathsf{out}}\big)$.*

[12] We assume that this is the "memoryless" version of ATL$^+$, with strategies represented as functions from *states* to choices.

Proposition 78 *The length of $tr_M(\varphi)$ is $l' = O(n + l)$, where l is the length of φ, and n is the number of states in M.*

The following nondeterministic algorithm can be used to model check formula $\langle\langle A\rangle\rangle\varphi$ of the "memoryless" ATL* in model M':

1. Recursively compute the interpretations of the state subformulae of φ in M' (e.g., for $\varphi \equiv \bigcirc\psi$, compute the set of states that satisfy ψ);
2. Guess the collective strategy S_A. Note that the size of S_A is $O(nkd)$;
3. "Trim" model M', removing all A's choices that do not appear in S_A. As M' is turn-based, the operation requires only $O(nkd)$ steps, and yields a turn-based ATS M'' with no more states and transitions than M';
4. Model-check CTL* formula $\mathsf{A}\varphi$ in M''.

Note that $\mathsf{A}\, next_M(\varphi) \Leftrightarrow \neg\mathsf{E}\bigvee_{q_i\notin[\![\varphi]\!]}(\overline{\mathsf{q_i}} \vee \overline{\mathsf{real}} \vee \overline{\mathsf{choice}})\,\mathcal{U}\ \overline{\mathsf{out}} \Leftrightarrow \neg\bigvee_{q_i\notin[\![\varphi]\!]}\mathsf{E}(\overline{\mathsf{q_i}} \vee \overline{\mathsf{real}} \vee \overline{\mathsf{choice}})\,\mathcal{U}\ \overline{\mathsf{out}}$, which is a formula of "vanilla" CTL, and can be model-checked in deterministic polynomial time.[13] Note also that an array of strategies for all the cooperation modalities occurring in a complex "positive" formula can be guessed *before* the translation of the formula (as strategy $s_a(q) = \alpha$ in M transformed to an equivalent strategy $s'_a(q^a) = \{\{q^{a,\alpha}\}\}$ in M'). The size of the witness is still $O(nkdl)$, which gives us the following.

Corollary 8 *Model checking of an ATL formula φ in an ATS M is in $\boldsymbol{\Delta}_2^{\mathrm{P}}$ with respect to n, k, d, l. Model checking "Positive ATL" is even in* **NP**.

Thus, we obtained a proof of membership in $\boldsymbol{\Delta}_2^{\mathrm{P}}$ (resp. **NP**), alternative to the one in Section 6.4. We want to emphasize that the above algorithm is somewhat more general than the one in Section 6.4.1, because it does *not* employ "tightening" of the model. In principle, an equivalent tight model exists for each ATS if we consider alternating transition systems in isolation. However, the same does not have to hold when we extend ATL and ATS with additional modalities. For instance, for an ATL extension that handles imperfect information, we may want to require that a single strategy specifies identical choices in indistinguishable states (cf. [134]), which means that a choice must include all the states that are considered as possible outcomes by an agent in a given situation, and not only the ones that can *physically* occur [129]. In consequence, such a kind of alternating *epistemic* transition systems cannot be tight in most cases. The above algorithm is valid for all ATS, even for those which cannot be tightened in a given context.

6.6 Model Checking Strategic Abilities of Agents under Incomplete Information

In this section, we consider model checking of *ATL with imperfect* (or *incomplete*) *information*. Since no satisfying semantics based on alternating transition systems has been proposed so far for strategic abilities under imperfect information, we present our results for an extension of concurrent game structures only.

[13] We thank an anonymous reviewer of MFCS'05 for pointing this out.

Schobbens [121] proved that ATL$_{ir}$ model checking is intractable: more precisely, it is **NP**-hard and $\mathbf{\Delta_2^P}$-easy (i.e., can be solved through a polynomial number of calls to an oracle for some problem in **NP**) when the size of the model is defined in terms of the number of transitions. He also conjectured that the problem might be $\mathbf{\Delta_2^P}$-complete.

This section contains several new results. Firstly, we close the gap and prove that model checking ATL$_{ir}$ is $\mathbf{\Delta_2^P}$-hard, and hence indeed $\mathbf{\Delta_2^P}$-complete with respect to the number of transitions in the model and the length of the formula. The proof proceeds by a reduction of the **SNSAT** problem to ATL$_{ir}$ model checking, presented in Section 6.6.3. **NP** and $\mathbf{\Delta_2^P}$ are quite close, both belonging to the first level of the polynomial hierarchy, so our result might seem a minor one – although, technically, it was not that trivial to prove it. On the other hand, its importance goes well beyond model checking of ATL$_{ir}$. In fact, Theorem 38 yields immediate corollaries with $\mathbf{\Delta_2^P}$-completeness of other logics like ATOL [83], "Feasible ATEL" [86], CSL [73] etc., and $\mathbf{\Delta_2^P}$-hardness of ETSL [135].

We also point out that the problem is **NP**-complete for the "positive" sublanguage of ATL$_{ir}$.

Secondly, we show that the problem is $\mathbf{\Delta_3^P}$-complete in the number of states, agents and decisions in the model, and the length of the formula (and it is "only" $\mathbf{\Sigma_2^P}$-complete for "Positive ATL$_{ir}$"). Therefore, the problem sits in the same complexity class as model checking strategic abilities for *perfect* information games with respect to these parameters. We believe this is good news, as far as complexity is concerned, for agent logics dealing with imperfect information.

Finally, we point out that the difference between the perfect and imperfect information case lies in the modularity of strategies with respect to the property that the agents may want to enforce. For perfect information games, potential successfulness of sub-strategies is more independent and they can be computed (or guessed) incrementally, while imperfect information strategies refuse incremental analysis.

6.6.1 Existing Results

Model checking ATL$_{ir}$ has been proved to be **NP**-hard and $\mathbf{\Delta_2^P}$-easy in the number of transitions and the length of the formula [121]. Membership in $\mathbf{\Delta_2^P}$ was demonstrated through the following observation. If the formula to be model checked is of the form $\langle\langle A\rangle\rangle_{ir}\varphi$ (φ being $\bigcirc\psi$, $\Box\psi$ or $\psi_1\,\mathcal{U}\,\psi_2$), where φ contains no more cooperation modalities, then it is sufficient to guess a strategy for A, "trim" the model by removing all transitions that will never be executed (according to this strategy), and model check CTL formula Aφ in the resulting model. Thus, model checking an arbitrary ATL$_{ir}$ formula can be done by checking the subformulae iteratively, which requires a polynomial number of calls to an **NP** algorithm.[14]

NP-hardness follows from a reduction of the well known **SAT** problem. Here, we present a reduction which is somewhat different from the one in [121]. We will adapt it in Section 6.6.3 to prove $\mathbf{\Delta_2^P}$-hardness. In **SAT**, we are given

[14]The algorithm from [79] can be also used to demonstrate the upper bounds for the complexity of this problem.

a CNF formula $\varphi \equiv C_1 \wedge \cdots \wedge C_n$ involving k propositional variables from set $X = \{x_1, ..., x_k\}$. Each clause C_i can be written as $C_i \equiv x_1^{s_{i,1}} \vee \cdots \vee x_k^{s_{i,k}}$, where $s_{i,j} \in \{+, -, 0\}$; x_j^+ denotes a positive occurrence of x_j in C_i, x_j^- denotes an occurrence of $\neg x_j$ in C_i, and x_j^0 indicates that x_j does not occur in C_i. The problem asks if $\exists X.\varphi$, that is, if there is a valuation of $x_1, ..., x_k$ such that φ holds.

We construct the corresponding i-CGS M_φ as follows. There are two players: verifier $\mathbf{v}$ and refuter $\mathbf{r}$. The refuter decides at the beginning of the game which clause C_i will have to be satisfied: it is done by proceeding from the initial state q_0 to a "clause" state q_i. At q_i, verifier decides (by proceeding to a "proposition" state $q_{i,j}$) which of the literals $x_j^{s_{i,j}}$ from C_i will be attempted. Finally, at $q_{i,j}$, verifier attempts to prove C_i by declaring the underlying propositional variable x_j true (action $\top$) or false (action $\bot$). If she succeeds (i.e., if she executes $\top$ for x_j^+, or executes $\bot$ for x_j^-), then the system proceeds to the "winning" state $q_\top$. Otherwise, the system stays in $q_{i,j}$. Additionally, "proposition" states referring to the same variable are indistinguishable for verifier, so that she has to declare the same value of x_j in all of them within a uniform strategy. A sole ATL$_{ir}$ proposition yes holds only in the "winning" state $q_\top$. Obviously, states corresponding to literals x_j^0 can be omitted from the model.

Speaking more formally, $M_\varphi = \langle \mathbb{A}\mathrm{gt}, St, \Pi, \pi, Act, d, o, \sim_1, ..., \sim_k \rangle$, where:

- $\mathbb{A}\mathrm{gt} = \{\mathbf{v}, \mathbf{r}\}$,
- $St = \{q_0\} \cup St_{cl} \cup St_{prop} \cup \{q_\top\}$, where $St_{cl} = \{q_1, \ldots, q_n\}$, and $St_{prop} = \{q_{1,1}, \ldots, q_{1,k}, \ldots, q_{n,1}, \ldots, q_{n,k}\}$;
- $\Pi = \{\mathsf{yes}\}$, $\quad \pi(\mathsf{yes}) = \{q_\top\}$,
- $Act = \{1, ..., \max(k, n), \top, \bot\}$,
- $d(\mathbf{v}, q_0) = d(\mathbf{v}, q_\top) = \{1\}$, $\quad d(\mathbf{v}, q_i) = \{1, ..., k\}$,
  $d(\mathbf{v}, q_{i,j}) = \{\top, \bot\}$,
  $d(\mathbf{r}, q) = \{1, ..., n\}$ for $q = q_0$, and $d(\mathbf{r}, q) = \{1\}$ otherwise;
- $o(q_0, 1, i) = q_i$, $\quad o(q_i, j, 1) = q_{i,j}$,
  $o(q_{i,j}, \top, 1) = q_\top$ if $s_{i,j} = +$, and $q_{i,j}$ otherwise,
  $o(q_{i,j}, \bot, 1) = q_\top$ if $s_{i,j} = -$, and $q_{i,j}$ otherwise;
- $q_0 \sim_\mathbf{v} q$ iff $q = q_0$, $\quad q_i \sim_\mathbf{v} q$ iff $q = q_i$, $\quad q_{i,j} \sim_\mathbf{v} q$ iff $q = q_{i',j}$.

As an example, model M_φ for $\varphi \equiv (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$ is presented in Figure 6.11.

Theorem 35 *φ is satisfiable iff* $M_\varphi, q_0 \models \langle\langle \mathbf{v} \rangle\rangle_{ir} \Diamond \mathsf{yes}$.

Proof

($\Rightarrow$) Firstly, if there is a valuation that makes φ true, then for each clause C_i one can choose a literal out of C_i that is made true by the valuation. The choice, together with the valuation, corresponds to a uniform strategy for $\mathbf{v}$ such that, for all possible executions, $q_\top$ is achieved at the end.

($\Leftarrow$) Conversely, if $M_\varphi, q_0 \models \langle\langle \mathbf{v} \rangle\rangle_{ir} \Diamond \mathsf{yes}$, then there is a strategy $s_\mathbf{v}$ such that $q_\top$ is achieved for all paths from $out(q_0, s_\mathbf{v})$. But then the valuation, which assigns propositions $x_1, ..., x_k$ with the same values as $s_\mathbf{v}$, satisfies φ. ■

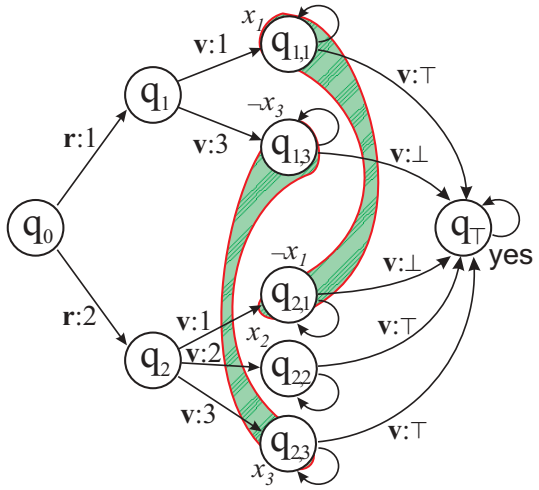


Figure 6.11: An i-CGS for checking satisfiability of $\varphi \equiv (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$

Both the number of states and transitions in M_φ are linear in the length of φ, and the construction of M requires linear time too. Thus, the model checking problem for ATL$_{ir}$ is **NP**-hard. Note that it is **NP**-hard even for formulae with a single cooperation modality, and turn-based models with at most two agents.[15]

6.6.2 NP-completeness for "Positive ATL"

We already investigated the complexity of ATL$_{ir}$ model checking in [79], concluding that the problem was **NP**-complete. Unfortunately, our claim was incorrect: the error occurred in the way we handled negation in our model checking algorithm (cf. [95]). Still, the algorithm from [79] *was* correct for "positive" formulae of ATL$_{ir}$: in this case, we can do the same trick as in Section 6.3.3, and guess all the relevant strategies beforehand. The size of the witness is still polynomial in this case: more precisely, it is $\mathbf{O}(ml)$, where m is the number of transitions, and l is the length of the formula. Thus, the following holds.

Theorem 36 *Model checking of "Positive ATL$_{ir}$" is* **NP***-complete with respect to the number of transitions in the model and the length of the formula.*

Proof A nondeterministic algorithm that checks formula φ in model M is presented in Figure 6.12. Calls to $mcheck_{CTL}$ refer to any established CTL model-checker (e.g. [30]). As for the time necessary to carry out the procedure: guessing the strategies can be done in time $\mathbf{O}(ml)$, while "trimming" the model, checking CTL formulae, and getting rid of the states in which agents may not know that the strategy is successful, can all be done in time $\mathbf{O}(m)$ (recursively for subformulae). Thus, the algorithm terminates in time

[15] In fact, it is **NP**-hard even for models with a single agent, although the construction must be a little different to demonstrate this.

function $mcheck_4(M, \varphi)$**;**
Returns the set of states in M, in which formula φ holds.

- assign cooperation modalities in φ with subsequent numbers $1, ..., c$;

  // note that $c \leq l$

  // we will denote the coalition from the ith cooperation modality in φ as $\varphi[i]$

- for each $i = 1, ..., c$, assign the agents in $\varphi[i]$ with numbers $1, ..., k_c$;

  // note that $k_c \leq k$ and $k_c \leq l$

  // we will denote the jth agent in A with $A[j]$

- guess an array $choice$ such that, for each $i = 1, ..., c$, $q \in St$, and $j = 1, ..., k_c$, we have that $choice[i][q][j] \in d_{\varphi[i][j]}(q)$, and for each $q' \in St$ such that $q \sim_{\varphi[i][j]} q'$ we have $choice[i][q][j] = choice[i][q'][j]$;

  // at this point, the optimal choices for all coalitions in φ are guessed

  // note that the size of $choice$ is $\mathbf{O}(ml)$

  // by $choice|_i$, we will denote the array $choice$ with rows $1, ..., i-1$ removed

- return $eval_4(M, \varphi, choice)$;

function $eval_4(M, \varphi, choice)$**;**
Returns the states in which φ holds, given choices for all the coalitions from φ.

case $\varphi \in \Pi$**:** return $\{q \mid \varphi \in \pi(q)\}$;
case $\varphi = \neg\psi$**:** return $Q \setminus eval_4(M, \psi, choice)$;
case $\varphi = \psi_1 \vee \psi_2$**:** return $eval_4(M, \psi_1, choice) \cup eval_4(M, \psi_2, choice)$;
case $\varphi = \langle\langle A\rangle\rangle T\psi$, where $T = \bigcirc$ or $\Box$**:**
  $Q_1 := eval_4(M, \psi, choice|_2)$; $M' := trim(M, choice[1])$;
  add to M' new proposition $\overline{\mathsf{p}}$ with $\pi(\overline{\mathsf{p}}) = Q_1$;
  $Q_2 := mcheck_{CTL}(M', \mathsf{A}T\,\overline{\mathsf{p}})$;
  return $\{q \in St \mid \forall a, q' \,.\, a \in A \wedge q \sim_a q' \Rightarrow q' \in Q_2\}$;
case $\varphi = \langle\langle A\rangle\rangle \psi_1 \,\mathcal{U}\, \psi_2$**:**
  c' := the number of cooperation modalities in ψ_1;
  $Q_1 := eval_4(M, \psi_1, choice|_2)$; $Q_2 := eval_4(M, \psi_2, choice|_{c'+2})$;
  $M' := trim(M, choice[1])$;
  add to M' new propositions $\overline{\mathsf{p}}_1, \overline{\mathsf{p}}_2$ with $\pi(\overline{\mathsf{p}}_1) = Q_1$, $\pi(\overline{\mathsf{p}}_2) = Q_2$;
  $Q_3 := mcheck_{CTL}(M', \mathsf{A}\overline{\mathsf{p}}_1 \,\mathcal{U}\, \overline{\mathsf{p}}_2)$;
  return $\{q \in St \mid \forall a, q' \,.\, a \in A \wedge q \sim_a q' \Rightarrow q' \in Q_3\}$;
end case

Figure 6.12: Nondeterministic algorithm for model checking "Positive ATL$_{ir}$".

$\mathbf{O}(ml)$. Combining it with the **NP**-hardness result from [121], we obtain the theorem. ■

function $trim(M, thischoice)$**;**
Returns the CTL model, which includes exactly the transitions that can occur
when A execute choices from $thischoice$.

- $\mathcal{R} := \varnothing$; `// the CTL transition relation (contains pairs of states)`
- for each $q \in St$ and tuple $resp$ of choices from $\mathbb{A}\mathrm{gt} \setminus A$, such that $resp[a] \in d(a, q)$:
  - $q' := o(q, thischoice[q], resp)$;
  - $\mathcal{R} := \mathcal{R} \cup \{\langle q, q' \rangle\}$;
- return $\langle St, \mathcal{R}, \Pi, \pi \rangle$;

Figure 6.13: Nondeterministic algorithm for "Positive ATL$_{ir}$", ctd.

Note that the exhaustive deterministic algorithm that checks all possible strategies runs in time $\mathbf{O}(nd^{kn}l) = \mathbf{O}(n(m/n)^n l)$, even for "Positive ATL$_{ir}$".

$\mathbf{\Delta_2^P}$-hardness for full ATL$_{ir}$ is proved in the next section.

6.6.3 Model Checking ATL$_{ir}$ Is Indeed $\mathbf{\Delta_2^P}$-complete

Let us first recall (after [95]) the definition of **SNSAT**, a typical $\mathbf{\Delta_2^P}$-hard problem.

Definition 49 (SNSAT)
Input: *p sets of propositional variables $X_r = \{x_{1,r}, ..., x_{k,r}\}$, p propositional variables z_r, and p Boolean formulae φ_r in CNF, with each φ_r involving only variables in $X_r \cup \{z_1, ..., z_{r-1}\}$, with the following requirement:*

$$z_r \equiv \textit{ there exists an assignment of variables in } X_r \textit{ such that } \varphi_r \textit{ is true.}$$

We will also write, by abuse of notation, $z_r \equiv \exists X_r\, \varphi_r(z_1, ..., z_{r-1}, X_r)$.
Output: *The truth-value of z_p (i.e., $\top$ or $\bot$).*

Let n be the maximal number of clauses in any $\varphi_1, ..., \varphi_p$ from the given input. Now, each φ_r can be written as:

$$\varphi_r \equiv C_1^r \wedge \cdots \wedge C_n^r, \textit{ and } C_i^r \equiv x_{1,r}^{s_{i,1}^r} \vee \cdots \vee x_{k,r}^{s_{i,k}^r} \vee z_1^{s_{i,k+1}^r} \vee \ldots z_{r-1}^{s_{i,k+r-1}^r}.$$

Again, $s_{i,j}^r \in \{+, -, 0\}$; x^+ denotes a positive occurrence of x, x^- denotes an occurrence of $\neg x$, and x^0 indicates that x does not occur in the clause. Similarly, $s_{i,k+j}^r$ defines the "sign" of z_j in clause C_i^r. Given such an instance of **SNSAT**, we construct a sequence of concurrent game structures M_r for $r = 1, ..., p$ in a similar way to the construction in Section 6.6.1. That is, clauses and variables $x_{i,r}$ are handled in exactly the same way as before. Moreover, if z_i occurs as a positive literal in φ_r, we embed M_i in M_r, and add a transition to the initial state q_0^i of M_i. If $\neg z_i$ occurs in φ_r, we do almost the

same: the only difference is that we split the transition into two steps, with a state neg_i^r (labeled with an ATL$_{ir}$ proposition neg) added in between.

More formally, $M_r = \langle \mathbb{A}\mathrm{gt}, St^r, \Pi, \pi^r, Act^r, d^r, o^r, \sim_1^r, ..., \sim_k^r \rangle$, where:

- $\mathbb{A}\mathrm{gt} = \{\mathbf{v}, \mathbf{r}\}$,
- $St^r = \{q_0^r, q_1^r, \ldots, q_n^r, q_{1,1}^r, \ldots, q_{n,k}^r, neg_1^r, \ldots, neg_{r-1}^r, q_\top\} \cup St^{r-1}$,
- $\Pi = \{\mathsf{yes}, \mathsf{neg}\}$, $\pi^r(\mathsf{yes}) = \{q_\top\}$, $\pi^r(\mathsf{neg}) = \{neg_i^j \mid i, j = 1, ..., r\}$,
- $Act^r = \{1, ..., \max(k + r - 1, n), \top, \bot\}$,
- $d^r(\mathbf{v}, q_0^r) = d^r(\mathbf{v}, neg_i^r) = d^r(\mathbf{v}, q_\top) = \{1\}$, $d^r(\mathbf{v}, q_i^r) = \{1, ..., k + r - 1\}$, $d^r(\mathbf{v}, q_{i,j}^r) = \{\top, \bot\}$,
  $d^r(\mathbf{r}, q) = \{1, ..., n\}$ for $q = q_0^r$ and $\{1\}$ for the other $q \in St^r$.
  For $q \in St^{r-1}$, we simply include the function from M_{r-1}: $d^r(a, q) = d^{r-1}(a, q)$;
- $o^r(q_0^r, 1, i) = q_i^r$, $o^r(q_i^r, j, 1) = q_{i,j}^r$ for $j \leq k$,
  $o^r(q_i^r, k + j, 1) = q_0^j$ if $s_{i,k+j}^r = +$, and $o^r(q_i^r, k + j, 1) = neg_j^r$ if $s_{i,k+j}^r = -$,
  $o^r(neg_j^r, 1, 1) = q_0^j$,
  $o^r(q_{i,j}^r, \top, 1) = q_\top$ if $s_{i,j}^r = +$, and $q_{i,j}^r$ otherwise,
  $o^r(q_{i,j}^r, \bot, 1) = q_\top$ if $s_{i,j}^r = -$, and $q_{i,j}^r$ otherwise.
  For $q \in St^{r-1}$, we include the transitions from M_{r-1}: $o^r(q, \alpha) = o^{r-1}(q, \alpha)$;
- $q_0^r \sim_\mathbf{v} q$ iff $q = q_0^r$, $q_i^r \sim_\mathbf{v} q$ iff $q = q_i^r$, $q_{i,j}^r \sim_\mathbf{v} q$ iff $q = q_{i',j}^r$.
  For $q, q' \in St^{r-1}$, we include the tuples from M_{r-1}: $q \sim_\mathbf{v}^r q'$ iff $q \sim_\mathbf{v}^{r-1} q'$.

As an example, model M_3 for $\varphi_3 \equiv (x_3 \vee \neg z_2) \wedge (\neg x_3 \vee \neg z_1)$, $\varphi_2 \equiv z_1 \wedge \neg z_1$, $\varphi_1 \equiv (x_1 \vee x_2) \wedge \neg x_1$, is presented in Figure 6.14.

Theorem 37 *Let*

$$\begin{aligned} \Phi_1 &\equiv \langle\langle \mathbf{v} \rangle\rangle_{ir} (\neg \mathsf{neg}) \,\mathcal{U}\, \mathsf{yes}, \\ \Phi_i &\equiv \langle\langle \mathbf{v} \rangle\rangle_{ir} (\neg \mathsf{neg}) \,\mathcal{U}\, (\mathsf{yes} \vee (\mathsf{neg} \wedge \mathsf{A}\bigcirc \neg \Phi_{i-1})). \end{aligned}$$

Now, for all r: z_r is true iff $M_r, q_0^r \models \Phi_r$.

Before we prove the theorem, we state an important lemma. Lemma 19 says that "overlong" formulae Φ_i do not introduce new properties of model M_r. More precisely, a formula Φ_i that includes more "nestings" than model M_r can be as well reduced to Φ_{i-1} when model checked in M_r, q_0^r.

Lemma 19 *For $i \geq r$: $M_r, q_0^r \models \Phi_i$ iff $M_r, q_0^r \models \Phi_{i+1}$.*

Proof (induction on r)

1. For $r = 1$: $M_1, q_0^1 \models \Phi_i$ iff $M_1, q_0^1 \models \langle\langle \mathbf{v} \rangle\rangle_{ir} \Diamond \mathsf{yes}$ iff $M_1, q_0^1 \models \Phi_{i+1}$, because M_1 does not include states that satisfy neg.

2. For $r > 1$: $M_r, q_0^r \models \Phi_{i+1} \equiv \langle\langle \mathbf{v} \rangle\rangle_{ir} (\neg \mathsf{neg}) \,\mathcal{U}\, (\mathsf{yes} \vee (\mathsf{neg} \wedge \mathsf{A}\bigcirc \neg \Phi_i))$ iff $\exists s_\mathbf{v} \forall \lambda \in out(q_0^r, s_\mathbf{v}) \exists u \forall w \leq u. \big((M_r, \lambda[u] \models \mathsf{yes}$ *or* $M_r, \lambda[u] \models \mathsf{neg} \wedge \mathsf{A}\bigcirc \neg \Phi_i)$ *and* $(M_r, \lambda[w] \models \neg \mathsf{neg})\big)$. [*]
   However, each state satisfying neg has exactly one outgoing transition, so $M_r, \lambda[u] \models \mathsf{neg} \wedge \mathsf{A}\bigcirc \neg \Phi_i$ is equivalent to $M_r, \lambda[u] \models \mathsf{neg}$ *and* $M_r, \lambda[u+$

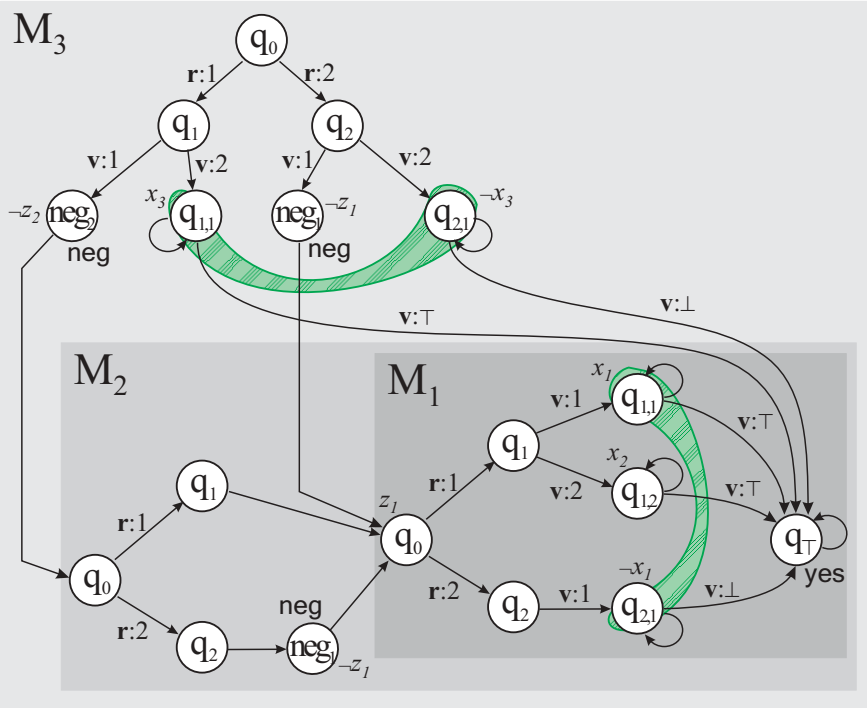


Figure 6.14: An i-CGS for the reduction of **SNSAT**. The superscripts in state labels are omitted since they can be deduced from the sub-machine in which the state resides.

1] $\models \neg\Phi_i$. Thus, [*] iff $\exists s_{\mathbf{v}} \forall \lambda \in out(q_0^r, s_{\mathbf{v}}) \exists u \forall w \leq u.\big((M_r, \lambda[u] \models \mathsf{yes}$ *or* $M_r, \lambda[u] \models \mathsf{neg}$ *and* $M_r, \lambda[u+1] \models \neg\Phi_i)$ *and* $(M_r, \lambda[w] \models \neg\mathsf{neg})\big)$ [**].

Note that, by the construction of M_r, $\lambda[u+1]$ must refer to the initial state q_0^j of some "submodel" M_j, $j < r \leq i$. Thus, $M_r, \lambda[u+1] \models \neg\Phi_i$ iff $M_j, q_0^j \models \neg\Phi_i$ iff (by induction) $M_j, q_0^j \models \neg\Phi_{i-1}$ iff $M_j, \lambda[u+1] \models \neg\Phi_{i-1}$.

So, [**] iff $\exists s_{\mathbf{v}} \forall \lambda \in out(q_0^r, s_{\mathbf{v}}) \exists u \forall w \leq u.\big((M_r, \lambda[u] \models \mathsf{yes}$ *or* $M_r, \lambda[u] \models \mathsf{neg}$ *and* $M_r, \lambda[u+1] \models \neg\Phi_{i-1})$ *and* $(M_r, \lambda[w] \models \neg\mathsf{neg})\big)$ iff $M_r, q_0^r \models \langle\langle \mathbf{v} \rangle\rangle_{ir} (\neg\mathsf{neg})\, \mathcal{U}\, (\mathsf{yes} \vee (\mathsf{neg} \wedge \mathsf{A}\bigcirc \neg\Phi_{i-1})) \equiv \Phi_i$.

■

Proof of Theorem 37 Induction on r:

1. For $r = 1$: we use the proof of Theorem 35.
2. For $r > 1$:

   For the implication from left to right (⇒): let z_r be true: then, there is a valuation of X_r such that φ_r holds. We construct $s_{\mathbf{v}}$ as in the proof of Theorem 35. In case that some x_i^s has been "chosen" in clause C_i^r, we are done. In case that some z_j^- has been "chosen" in clause C_i^r

(note: j must be smaller than i), we have (by induction) that $M_j, q_0^j \models \neg\Phi_j$. By Lemma 19, also $M_j, q_0^j \models \neg\Phi_r$, and hence $M_r, q_0^j \models \neg\Phi_r$. So we can make the same choice (i.e., z_j^-) in $s_{\mathbf{v}}$, and this will lead to state neg_j^r, in which it holds that $\mathsf{neg} \wedge \mathsf{A}\bigcirc\neg\Phi_r$.

In case that some z_j^+ has been "chosen" in clause C_i^r, we have (by induction) that $M_j, q_0^j \models \Phi_j$, and hence, by Lemma 19, $M_j, q_0^j \models \Phi_r$. That is, there is a strategy $s'_{\mathbf{v}}$ in M_j such that $(\neg\mathsf{neg})\,\mathcal{U}\,(\mathsf{yes} \vee (\mathsf{neg} \wedge \mathsf{A}\bigcirc\neg\Phi_{r-1}))$ holds for all paths from $out(q_0^j, s'_{\mathbf{v}})$. As the states in M_j have no epistemic links to states outside of it, we can merge $s'_{\mathbf{v}}$ into $s_{\mathbf{v}}$.

For the other direction ($\Leftarrow$): let $M_r, q_0^r \models \Phi_r \equiv \langle\langle \mathbf{v} \rangle\rangle_{ir} (\neg\mathsf{neg})\,\mathcal{U}\,(\mathsf{yes} \vee (\mathsf{neg} \wedge \mathsf{A}\bigcirc\neg\Phi_{r-1}))$. We take the strategy $s_{\mathbf{v}}$ that enforces $(\neg\mathsf{neg})\,\mathcal{U}\,(\mathsf{yes} \vee (\mathsf{neg} \wedge \mathsf{A}\bigcirc\neg\Phi_{r-1}))$. We first consider the clause C_i^r for which a "propositional" state is chosen by $s_{\mathbf{v}}$. The strategy defines a uniform valuation for X_r that satisfies these clauses. For the other clauses, we have two possibilities:

- $s_{\mathbf{v}}$ chooses q_0^j in the state corresponding to C_i^r. Neither yes nor neg have been encountered on this path yet, so we can take $s_{\mathbf{v}}$ to demonstrate that $M_r, q_0^j \models \Phi_r$, and hence $M_j, q_0^j \models \Phi_r$. By Lemma 19, also $M_j, q_0^j \models \Phi_j$. By induction, z_j must be true, and hence clause C_i^r is satisfied.
- $s_{\mathbf{v}}$ chooses neg_j^r in the state corresponding to C_i^r. Then, it must be that $M_r, neg_j^r \models \mathsf{A}\bigcirc\neg\Phi_{r-1}$, and hence $M_j, q_0^j \models \neg\Phi_{r-1}$. By Lemma 19, also $M_j, q_0^j \models \neg\Phi_j$. By induction, z_j must be false, and hence clause C_i^r (containing $\neg z_j$) is also satisfied.

■

Thus, in order to determine the value of z_p, it is sufficient to model check Φ_p in M_p, q_0^p. Note that model M_p consists of $O(|\varphi|p)$ states and $O(|\varphi|p)$ transitions, where $|\varphi|$ is the maximal length of formulae $\varphi_1, ..., \varphi_p$. Moreover, the length of formula Φ_p is linear in p, and the construction of M_p and Φ_p can be also done in time $O(|\varphi|p)$ and $O(p)$, respectively. In consequence, we obtain a polynomial reduction of **SNSAT** to ATL$_{ir}$ model checking.

Theorem 38 *Model checking ATL$_{ir}$ is $\mathbf{\Delta_2^P}$-complete with respect to the number of transitions in the model, and the length of the formula. The problem is $\mathbf{\Delta_2^P}$-complete even for turn-based models with at most two agents.*

6.6.4 The Complexity Refined

One of the problems with model checking formulae of ATL is that the number of transitions m in a model is not bounded by n^2, and can be very large: more precisely, $m = \mathbf{O}(nd^k)$ where n is the number of states, k the number of agents, and d the maximal number of decisions per agent per state. Thus, m is exponential in k unless the model is turn-based or the number of agents is fixed. In consequence, ATL model checking becomes $\mathbf{\Delta_3^P}$-complete when n, k, d, l are the parameters of the problem, and model checking "Positive ATL" becomes $\mathbf{\Sigma_2^P}$-complete. In this section, we show that ATL$_{ir}$ model

checking is also $\mathbf{\Delta_3^P}$-complete (resp. $\mathbf{\Sigma_2^P}$-complete "Positive ATL$_{ir}$") in the same setting. To prove the lower bounds, it suffices to point out that:

Lemma 20 *ATL is semantically subsumed by ATL$_{ir}$. "Positive ATL" is semantically subsumed by "Positive ATL$_{ir}$".*

Proof In order to transform a concurrent game structure M to a corresponding imperfect information concurrent game structure M', we fix the indistinguishability relations as the minimal total reflexive relations, (i.e. $\sim_a = \{\langle q, q\rangle \mid q \in St\}$ for all $a \in \mathbb{A}\mathrm{gt}$), which means that the agents can distinguish between any two states. Let φ be a formula of ATL, and φ' the result of adding subscript ir in each cooperation modality in φ. Then, $M, q \models \varphi$ iff $M', q \models \varphi'$. Thus, ATL (resp. "Positive ATL") model checking can be seen as a special case of ATL$_{ir}$ (resp. "Positive ATL$_{ir}$") model checking. ∎

To show that the problem is $\mathbf{\Sigma_2^P}$-easy for "Positive ATL$_{ir}$", we present a refinement of the algorithm from Section 6.6.2 in Figures 6.15 and 6.16.

Proposition 79 *Function $mcheck_5$ defines a nondeterministic Turing machine that runs in time $\mathbf{O}(n^2kl)$, making calls to an $\mathbf{NP}$ oracle. The oracle itself is a nondeterministic Turing machine that runs in time $\mathbf{O}(n+k)$. The size of witnesses is never more than $\mathbf{O}(nkl)$.*

Proof The main idea is as follows. Firstly, we guess nondeterministically *all* the strategies for the cooperation modalities that occur in formula φ (we do it beforehand, as in Section 6.6.2). The strategies must be uniform, so setting $s_a(q)$ fixes automatically $s_a(q')$ for all $q \sim_a q'$. Then we model check φ recursively: for each subformula $\langle\!\langle A\rangle\!\rangle_{ir}\psi$, we assume the respective strategy and check the formula $\langle\!\langle\varnothing\rangle\!\rangle_{ir}\psi$. To do so, we take ATL formula $\langle\!\langle A\rangle\!\rangle\psi$ as input, and employ the standard ATL model checking algorithm from [8] with one important modification: each time function $pre(A, Q_1)$ is called, it assumes the respective A's choices, and checks whether $q \in pre(A, Q_1)$ by calling an $\mathbf{NP}$ oracle (*"is there a response from the opposition in q that leads to a state outside Q_1?"*) and reversing its answer. Note that the latter amounts to checking $M', q \models \langle\!\langle\varnothing\rangle\!\rangle\bigcirc \mathbf{Q_1}$, where M' is model M with A's actions fixed accordingly, and $\mathbf{Q_1}$ is a new proposition that holds exactly in states Q_1. Finally, we get rid of the states that have indistinguishable counterparts for which the assumed strategy is not successful. Note that, in the middle part of the algorithm, we use an adaptation of the ATL model checking procedure, which *iterates* over states of the system. This kind of iterative solution is possible because $\langle\!\langle\varnothing\rangle\!\rangle_{ir}\psi \equiv \langle\!\langle\varnothing\rangle\!\rangle\psi$ (although, of course, the analogous property does not hold for $\langle\!\langle A\rangle\!\rangle_{ir}$ in general).

The detailed algorithm is shown in Figures 6.15 and 6.16. The procedure is very similar to the "Positive ATL" model checking algorithm from Section 6.3.3. Analogous complexity analysis applies: first, the number of iterations *within* one single call of function $eval$, as well as the number of calls to pre, is $\mathbf{O}(n)$; next, function pre runs in $\mathbf{O}(n)$ steps, including calls to the oracle; removing the states for which a member of the coalition can have any doubts can be done in time $\mathbf{O}(n^2k)$; finally, $eval$ is called at most $\mathbf{O}(l)$ times. In consequence, we get a nondeterministic polynomial algorithm that makes calls to an $\mathbf{NP}$ oracle. ∎

function $mcheck_5(M, \varphi)$**;**
Returns the set of states in M, in which formula φ holds.

- assign cooperation modalities in φ with subsequent numbers $1, ..., c$;

  // note that $c \leq l$

  // we will denote the coalition from the ith cooperation modality in φ as $\varphi[i]$

- for each $i = 1, ..., c$, assign the agents in $\varphi[i]$ with numbers $1, ..., k_c$;

  // note that $k_c \leq k$ and $k_c \leq l$

  // we will denote the jth agent in A with $A[j]$

- guess an array $choice$ such that, for each $i = 1, ..., c$, $q \in St$, and $j = 1, ..., k_c$, we have that $choice[i][q][j] \in d_{\varphi[i][j]}(q)$, and for each $q' \in St$ such that $q \sim_{\varphi[i][j]} q'$ we have $choice[i][q][j] = choice[i][q'][j]$;

  // at this point, the optimal choices for all coalitions in φ are guessed

  // note that the size of $choice$ is $\mathbf{O}(nkl)$

  // by $choice|_i$, we will denote the array $choice$ with rows $1, ..., i-1$ removed

- return $eval_5(M, \varphi, choice)$;

function $eval_5(M, \varphi, choice)$**;**
Returns the states in which φ holds, given choices for all the coalitions from φ.

case $\varphi \in \Pi$ **:** return $\{q \mid \varphi \in \pi(q)\}$;
case $\varphi = \neg\psi$ **:** return $Q \setminus eval_5(M, \psi, choice)$;
case $\varphi = \psi_1 \vee \psi_2$ **:** return $eval_5(M, \psi_1, choice) \cup eval_5(M, \psi_2, choice)$;
case $\varphi = \langle\langle A\rangle\rangle \bigcirc \psi$ **:**
  $Q_1 := pre_5(A, eval_5(M, \psi, choice|_2), M, choice[1])$;
  return $\{q \in St \mid \forall a, q' \,.\, a \in A \wedge q \sim_a q' \Rightarrow q' \in Q_1\}$;
case $\varphi = \langle\langle A\rangle\rangle \Box \psi$ **:** $Q_1 := St$; $Q_2 := Q_3 := eval_5(M, \psi, choice|_2)$;
  while $Q_1 \not\subseteq Q_2$ **do** $Q_1 := Q_1 \cap Q_2$; $Q_2 := pre_5(A, Q_1, M, choice[1]) \cap Q_3$
od;
  return $\{q \in St \mid \forall a, q' \,.\, a \in A \wedge q \sim_a q' \Rightarrow q' \in Q_1\}$;
case $\varphi = \langle\langle A\rangle\rangle \psi_1 \,\mathcal{U}\, \psi_2$ **:** $c' :=$ the number of cooperation modalities in ψ_1;
  $Q_1 := \varnothing$; $Q_2 := eval_5(M, \psi_1, choice|_2)$; $Q_3 := eval_5(M, \psi_2, choice|_{c'+2})$;
  while $Q_3 \not\subseteq Q_1$ **do** $Q_1 := Q_1 \cup Q_3$; $Q_3 := pre_5(A, Q_1, M, choice[1]) \cap Q_2$
od;
  return $\{q \in St \mid \forall a, q' \,.\, a \in A \wedge q \sim_a q' \Rightarrow q' \in Q_1\}$;
end case

Figure 6.15: The model checking algorithm refined (main part).

Like for ATL, the algorithm can be easily adapted to handle arbitrary ATL$_{ir}$ formulae in time $\mathbf{\Delta_3^P}$ (strategies are guessed for each $\langle\langle A\rangle\rangle_{ir}$ separately, and not in advance). Thus, we get the following.

function $pre_5(A, Q_1, M, thischoice)$**;**
Returns the set of states, for which the A's choices from $thischoice$ enforce that the next state is in Q_1, regardless of what agents from $\mathbb{A}\mathrm{gt} \setminus A$ do.

- $Q_2 := \varnothing$;
- for each $q \in St$: **if** $oracle_5(A, Q_1, M, thischoice, q) = yes$ **then** $Q_2 := Q_2 \cup \{q\}$ **fi**;
- return Q_2;

function $oracle_5(A, Q_1, M, thischoice, q)$**;**
Returns yes if, and only if, the A's choices from $thischoice$ in q enforce that the next state is in Q_1, regardless of what agents from $\mathbb{A}\mathrm{gt} \setminus A$ do.

- guess an array $resp$ such that, for each $a \in \mathbb{A}\mathrm{gt} \setminus A$, we have $resp[a] \in d(a, q)$;

  `// at this point, the most dangerous response from the opposition is guessed`

  `// note that the size of` ***resp*** `is` $\mathbf{O}(k)$
- **if** $o(q, thischoice[q], resp) \in Q_1$ **then** return yes **else** return no **fi**;

Figure 6.16: The model checking algorithm refined: pre-image and oracle.

Theorem 39 *Model checking ATL_{ir} formulae over i-CGS is $\mathbf{\Delta_3^P}$-complete with respect to the number of states, decisions and agents, and the length of formulae. Model checking "Positive ATL_{ir}" is $\mathbf{\Sigma_2^P}$-complete.*

6.6.5 Discussion

The result has been somewhat surprising to us, since it turns out that a *fine grained* analysis puts checking strategic abilities of agents under imperfect information in the same complexity class as for perfect information games. It is surprising because the first case appears to be *strictly* harder than the latter when we approach it from a more "distant" perspective (i.e. when the input parameters are less detailed).

Let us recall from Section 6.3 that the hardness of model checking ATL is due to simultaneous actions of agents, and can be demonstrated even for scenarios that consist of a single step. It turns out that restricting agents' strategies to uniform strategies only does not increase model checking complexity *enough* to shift it to a higher complexity class. Even the size of witnesses is the same in both cases.

What is different then, that makes model checking of ATL_{ir} harder than ATL in relation to the number of transitions?

Definitely *not* the number of transitions itself, because CGS can be seen as a special case of i-CGS. Comparison of model checking complexity for turn-based structures[16] can give us a hint in this respect. Note that, for such

[16] I.e., structures in which at each state there is a single agent who decides upon the next

structures, $m = \mathbf{O}(nd)$ and we can use the model checking algorithms from Section 6.6.2 and from [8] to model-check formulae of ATL$_{ir}$ and ATL, respectively.

Proposition 80 *Model checking ATL$_{ir}$ over turn-based i-CGS is $\mathbf{\Delta_2^P}$-complete (**NP**-complete for "Positive ATL$_{ir}$"), while model checking ATL over turn-based CGS can be done deterministically in time $\mathbf{O}(ndl)$. Since $d \leq n$ for turn-based structures, the latter bound can be replaced by $\mathbf{O}(n^2 l)$.*

The result can be generalised to systems in which only a fixed (or bounded) number of agents is acting in each state; we propose to call such systems *semi-turn-based* concurrent game structures. Note that systems with a fixed (or bounded) number of agents are a special case of semi-turn-based CGS.

Proposition 81 *Model checking ATL$_{ir}$ over semi-turn-based i-CGS is $\mathbf{\Delta_2^P}$-complete (**NP**-complete for "Positive ATL$_{ir}$"), while model checking ATL over semi-turn-based CGS can be done deterministically in time $\mathbf{O}(n^2 l)$.*

Moreover, the exhaustive model checking of ATL formulae can be done in time $\mathbf{O}(nd^k l)$, while, for ATL$_{ir}$ formulae, it can be done in $\mathbf{O}(nd^{kn} l)$ steps. This is due to the fact that $\langle\langle A\rangle\rangle \Box\varphi \equiv \varphi \wedge \langle\langle A\rangle\rangle \bigcirc \langle\langle A\rangle\rangle \Box\varphi$ and $\langle\langle A\rangle\rangle \varphi \mathcal{U} \psi \equiv \psi \vee \varphi \wedge \langle\langle A\rangle\rangle \bigcirc \langle\langle A\rangle\rangle \varphi \mathcal{U} \varphi$ in ATL, whereas analogous fixpoint characterisations do not hold for ATL$_{ir}$ modalities. Thus, successful ATL strategies can be computed incrementally, state by state. By contrast, uniform strategies must be considered *as a whole*, which requires much more backtracking if we check the possibilities exhaustively.

Nevertheless, we believe that the results in this section indicate that agent logics with imperfect information might not be unfeasible. If ATL formulae can be feasibly model-checked then agents with imperfect information are not *that* far away. And there already exist running model-checkers for ATL [9, 4], based on OBDD (Ordered Binary Decision Diagrams). Also, new model checking techniques, based on the idea of *Unbounded Model Checking*, are under development [87].

6.7 Conclusions

In this article, we discussed model checking complexity for several variants of alternating-time temporal logic ATL. We analyzed the complexity of model checking for explicit models when the size of models is defined in terms of states rather than transitions, and the number of agents is considered a parameter of the problem. Most importantly, we proved that the problem is intractable for all studied variants of the logic. First of all, we showed that model checking "Positive ATL" (i.e., ATL where the use of negation is restricted to literals) over concurrent game structures is $\mathbf{\Sigma_2^P}$-complete. Moreover, for the previous semantics based on alternating transition systems, the problem is "only" **NP**-complete. Using our results, Laroussinie, Markey and Oreiby proved subsequently in [95] that model checking of full ATL is $\mathbf{\Delta_3^P}$-complete for CGS and $\mathbf{\Delta_2^P}$-complete for ATS. All these results suggest that

transition; this can be modeled by requiring that $d(a, q)$ is a singleton for all but one agent.

		m, l	n, k, l	n_{local}, k, l
CTL		P [30]	P [30]	PSPACE [92]
P-ATL	ATS	P [7]	**NP** (Sect. 6.4)	EXPTIME [131]
	NATS	P (Sect. 6.4.5)	**NP** (Sect. 6.4.5)	
	CGS	P [8]	$\mathbf{\Sigma_2^P}$ (Sect. 6.3)	
ATL	ATS	P [7]	$\mathbf{\Delta_2^P}$ [95]	
	NATS	P (Sect. 6.4.5)	$\mathbf{\Delta_2^P}$ (Sect. 6.4.5)	
	CGS	P [8]	$\mathbf{\Delta_3^P}$ [95]	
P-ATL$_{ir}$		NP ([121] & Sec. 6.6.2)	$\mathbf{\Sigma_2^P}$ (Sect. 6.6.4)	?
ATL$_{ir}$		$\mathbf{\Delta_2^P}$ ([121] & Sec. 6.6.3)	$\mathbf{\Delta_3^P}$ (Sect. 6.6.4)	

Figure 6.17: Model checking complexity: completeness results for various settings of input parameters. Symbols n, k, m stand for the number od states, agents and transitions in the explicit model, l is the length of the formula, and n_{local} is the number of local states in a concurrent program. P-ATL stands for "Positive ATL", and P-ATL$_{ir}$ for "Positive ATL$_{ir}$".

using ATS may have some advantage over CGS. Secondly, we showed that ATL model checking over the broader class of nondeterministic alternating transition systems is still **NP**-complete for the "positive" formulae (and $\mathbf{\Delta_2^P}$-complete in the general case), and hence the ATS-based semantics might perhaps be used in a more convenient way than until now.

Finally, we proved that:

1. Model checking ATL$_{ir}$ (i.e., ATL with imperfect information) is $\mathbf{\Delta_2^P}$-complete in the number of transitions and the length of the formula (therefore closing a gap in existing research);
2. Model checking "Positive ATL$_{ir}$" is **NP**-complete in the number of transitions and the length of the formula;
3. Model checking ATL$_{ir}$ is $\mathbf{\Delta_3^P}$-complete when the size of models is defined in terms of states rather than transitions;
4. Model checking "Positive ATL$_{ir}$" is $\mathbf{\Sigma_2^P}$-complete in the same setting.

Thus, checking strategic ability under imperfect information falls in the same complexity class as checking strategic ability for perfect information agents, when a more refined analysis is conducted – which we consider somewhat surprising. We summarise the existing results on model checking ATL-related logics in Figure 6.7.

Additionally, we presented a truth-preserving translation of ATL models and formulae. The resulting models are always *turn-based*, which usually means an exponential decrease in the number of transitions. As turn-based alternating transition systems are very close to CTL models, as well as extensive form games with perfect information, one may hope that some interesting techniques can be transferred from CTL model checking and/or game theory this way. Moreover, our translation of models is independent from the translation of formulae, which allows for "pre-compiling" models when one wants to check various properties of a particular multi-agent system.

We would like to thank: Thomas Eiter for his help in the **NP**-hardness proof of *Sfc-SAT*; Rafał Somla for the discussions about model checking ATL and its various extensions; Nils Bulling for checking our checking of "weak until", and for checking our proof of Theorem 37; Pierre-Yves Schobbens for comments; last, but not least, the anonymous reviewers of MFCS'05, CEEMAS'05, ICTCS'05, BNAIC'05, and TOCS for their helpful remarks.

Chapter 7

Modular Interpreted Systems (joint work with Thomas Ågotnes)

Abstract. We propose a new class of representations that can be used for modeling (and model checking) temporal, strategic and epistemic properties of agents and their teams. Our representations borrow the main ideas from *interpreted systems* of Halpern, Fagin et al.; however, they are also modular and compact in the way *concurrent programs* are. We also mention preliminary results on model checking alternating-time temporal logic for this natural class of models.

`Keywords:` open computational systems, temporal and strategic logics, modeling methodology, model checking

7.1 Introduction

The logical foundations of multi-agent systems have received much attention in recent years. Logic has been used to represent and reason about, e.g., knowledge [42], time [39], cooperation and strategic ability [8]. Lately, an increasing amount of research has focused on higher level representation languages for models of such logics, motivated mainly by the need for compact representations, and for representations that correspond more closely to the actual systems which are modeled. Multi-agent systems are *open* systems, in the sense that agents interact with an environment only partially known in advance. Thus, we need representations of models of multi-agent systems which are *modular*, in the sense that a component, such as an agent, can be replaced, removed, or added, without major changes to the representation of the whole model. However, as we argue in this paper, few existing representation languages are both modular, compact and computationally grounded on the one hand, and allow for representing properties of both knowledge and strategic ability, on the other.

In this paper we present a new class of representations for models of open

multi-agent systems, which are modular, compact and come with an implicit methodology for modeling and designing actual systems.

The structure of the paper is as follows. First, in Section 7.2, we present the background of our work – that is, logics that combine time, knowledge, and strategies. More precisely: modal logics that combine branching time, knowledge, and strategies under incomplete information. We start with computation tree logic CTL, then we add knowledge (CTLK), and then we discuss two variants of alternating-time temporal logic (ATL): one for the perfect, and one for the imperfect information case. The semantics of logics like the ones presented in Section 7.2 are usually defined over *explicit models* (Kripke structures) that enumerate all possible (global) states of the system. However, enumerating these states is one of the things one mostly wants to avoid, because there are too many of them even for simple systems. Thus, we usually need representations that are more *compact*. Another reason for using a more specialized class of models is that general Kripke structures do not always give enough help in terms of methodology, both at the stage of design, nor at implementation. This calls for a semantics which is more *grounded*, in the sense that the correspondence between elements of the model, and the entities that are modeled, is more immediate. In Section 7.3, we present an overview of representations that have been used for modeling and model checking systems in which time, action (and possibly knowledge) are important; we mention especially representations used for theoretical analysis. We point out that the compact and/or grounded representations of temporal models do not play their role in a satisfactory way when agents' strategies are considered. Finally, in Section 7.4, we present our framework of *modular interpreted systems* (MIS), and show where it fits in the picture. We conclude with a somewhat surprising hypothesis, that model checking ability under imperfect information for MIS can be computationally cheaper than model checking perfect information. Until now, almost all complexity results were distinctly in favor of perfect information strategies (and the others were indifferent).

7.2 Logics of Time, Knowledge, and Strategic Ability

First, we present the logics CTL, CTLK, ATL and ATL$_{ir}$ that are the starting point of our study.

7.2.1 Branching Time: CTL

Computation tree logic CTL [39] includes operators for temporal properties of systems: i.e., path quantifier E ("there is a path"), together with temporal operators: $\bigcirc$ ("in the next state"), $\Box$ ("always from now on") and $\mathcal{U}$ ("until").[1] Every occurrence of a temporal operator is immediately preceded by exactly one path quantifier (this variant of the language is sometimes called "vanilla" CTL).

[1] Additional operators A ("for every path") and $\Diamond$ ("sometime in the future") are defined in the usual way.

Let Π be a set of atomic propositions with a typical element p. CTL formulae φ are defined as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{E}\bigcirc\varphi \mid \mathsf{E}\Box\varphi \mid \mathsf{E}\varphi\,\mathcal{U}\,\varphi.$$

The semantics of CTL is based on Kripke models $M = \langle St, \mathcal{R}, \pi\rangle$, which include a nonempty set of states St, a state transition relation $\mathcal{R} \subseteq St \times St$, and a valuation of propositions $\pi : \Pi \to 2^{St}$. A *path* λ in M refers to a possible behavior (or computation) of system M, and can be represented as an infinite sequence of states $q_0q_1q_2...$ such that $q_i\mathcal{R}q_{i+1}$ for every $i = 0, 1, 2, ...$ We denote the ith state in λ by $\lambda[i]$. A *q-path* is a path that starts in q. Interpretation of a formula in a state q in model M is defined as follows:

$M, q \models p$ iff $q \in \pi(p)$;

$M, q \models \neg\varphi$ iff $M, q \not\models \varphi$;

$M, q \models \varphi \wedge \psi$ iff $M, q \models \varphi$ and $M, q \models \psi$;

$M, q \models \mathsf{E}\bigcirc\varphi$ iff there is a q-path λ such that $M, \lambda[1] \models \varphi$;

$M, q \models \mathsf{E}\Box\varphi$ iff there is a q-path λ such that $M, \lambda[i] \models \varphi$ for every $i \geq 0$;

$M, q \models \mathsf{E}\varphi\,\mathcal{U}\,\psi$ iff there is a q-path λ and $i \geq 0$ such that $M, \lambda[i] \models \psi$ and $M, \lambda[j] \models \varphi$ for every $0 \leq j < i$.

7.2.2 Adding Knowledge: CTLK

CTLK [115] is a straightforward combination of CTL and standard epistemic logic [55, 42]. Let $\mathbb{A}\mathrm{gt} = \{1, ..., k\}$ be a set of agents with a typical element a. Epistemic logic uses operators for representing agents' knowledge: $K_a\varphi$ is read as "agent a knows that φ". Models of CTLK extend models of CTL with epistemic indistinguishability relations $\sim_a \subseteq St \times St$ (one per agent). We assume that all $\sim_a$ are equivalences. The semantics of epistemic operators is defined as follows:

$M, q \models K_a\varphi$ iff $M, q \models \varphi$ for every q' such that $q \sim_a q'$.

Note that, when talking about agents' knowledge, we implicitly assume that agents may have imperfect information about the actual current state of the world (otherwise the notion of knowledge would be trivial). This does not have influence on the way we model evolution of a system as a single unit, but it will become important when particular agents and their strategies come to the fore.

7.2.3 Agents and Their Strategies: ATL

Alternating-time temporal logic ATL [8] is a logic for reasoning about temporal and strategic properties of open computational systems (multi-agent systems in particular). The language of ATL consists of the following formulae:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle\bigcirc\varphi \mid \langle\langle A\rangle\rangle\Box\varphi \mid \langle\langle A\rangle\rangle\varphi\,\mathcal{U}\,\varphi.$$

where $A \subseteq \mathbb{A}\mathrm{gt}$. Informally, $\langle\langle A\rangle\rangle\varphi$ says that agents A have a collective strategy to enforce φ. It should be noted that the CTL path quantifiers A, E can be expressed with $\langle\langle\varnothing\rangle\rangle, \langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle$ respectively.

The semantics of ATL is defined in so called *concurrent game structures (CGS s)*. A CGS is a tuple

$$M = \langle \mathbb{A}\mathrm{gt}, St, Act, d, o, \Pi, \pi \rangle,$$

consisting of: a set $\mathbb{A}\mathrm{gt} = \{1, \ldots, k\}$ of *agents*; set St of *states*; *valuation of propositions* $\pi : \Pi \to 2^{St}$; set Act of atomic *actions*. Function $d : \mathbb{A}\mathrm{gt} \times St \to 2^{Act}$ indicates the actions available to agent $a \in \mathbb{A}\mathrm{gt}$ in state $q \in St$. Finally, o is a deterministic *transition function* which maps a state $q \in St$ and an action profile $\langle \alpha_1, \ldots, \alpha_k \rangle \in Act^k, \alpha_i \in d(i, q)$, to another state $q' = o(q, \alpha_1, \ldots, \alpha_k)$.

Definition 50 *A (memoryless)* strategy *of agent a is a function $s_a : St \to Act$ such that $s_a(q) \in d(a, q)$.*[2] *A* collective strategy *S_A for a team $A \subseteq \mathbb{A}\mathrm{gt}$ specifies an individual strategy for each agent $a \in A$. Finally, the* outcome *of strategy S_A in state q is defined as the set of all computations that may result from executing S_A from q on:*

$out(q, S_A) = \{\lambda = q_0 q_1 q_2 ... \mid q_0 = q$ *and for each* $i = 1, 2, ...$ *there is* $\langle \alpha_1^{i-1}, ..., \alpha_k^{i-1} \rangle$ *such that* $\alpha_a^{i-1} = S_A(a)(q_{i-1})$ *for each* $a \in A$, $\alpha_a^{i-1} \in d(a, q_{i-1})$ *for each* $a \notin A$, *and* $o(q_{i-1}, \alpha_1^{i-1}, ..., \alpha_k^{i-1}) = q_i\}$.

The semantics of cooperation modalities is as follows:

- $M, q \models \langle\langle A \rangle\rangle \bigcirc \varphi$ iff there is a collective strategy S_A such that, for every $\lambda \in out(q, S_A)$, we have $M, \lambda[1] \models \varphi$;
- $M, q \models \langle\langle A \rangle\rangle \Box \varphi$ iff there exists S_A such that, for every $\lambda \in out(q, S_A)$, we have $M, \lambda[i]$ for every $i \geq 0$;
- $M, q \models \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$ iff there exists S_A such that for every $\lambda \in out(q, S_A)$ there is a $i \geq 0$, for which $M, \lambda[i] \models \psi$, and $M, \lambda[j] \models \varphi$ for every $0 \leq j < i$.

7.2.4 Agents with Imperfect Information: ATL$_{ir}$

As ATL does not include incomplete information in its scope, it can be seen as a logic for reasoning about agents who always have complete knowledge about the current state of the whole system. ATL$_{ir}$ [121] includes the same formulae as ATL, except that the cooperation modalities are presented with a subscript: $\langle\langle A \rangle\rangle_{ir}$ indicates that they address agents with imperfect *information* and imperfect *recall*. Formally, the recursive definition of ATL$_{ir}$ formulae is:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle_{ir} \bigcirc \varphi \mid \langle\langle A \rangle\rangle_{ir} \Box \varphi \mid \langle\langle A \rangle\rangle_{ir} \varphi \mathcal{U} \varphi$$

Models of ATL$_{ir}$, *concurrent epistemic game structures* (CEGS), can be defined as tuples $M = \langle \mathbb{A}\mathrm{gt}, St, Act, d, o, \sim_1, ..., \sim_k, \Pi, \pi \rangle$, where $\langle \mathbb{A}\mathrm{gt}, St, Act, d, o, \Pi, \pi \rangle$ is a CGS, and $\sim_1, ..., \sim_k$ are epistemic (equivalence) relations. It is required that agents have the same choices in indistinguishable states: $q \sim_a q'$ implies $d(a, q) = d(a, q')$. ATL$_{ir}$ restricts the strategies that can

[2] This is a deviation from the original semantics of ATL [8], where strategies assign agents' choices to *sequences* of states, which suggests that agents can by definition recall the whole history of each game. While the choice of one or another notion of strategy affects the semantics of the full ATL*, and most ATL extensions (e.g. for games with imperfect information), it should be pointed out that both types of strategies yield equivalent semantics for "pure" ATL (cf. [121]).

be used by agents to *uniform strategies*, i.e. functions $s_a : St \to Act$, such that: (1) $s_a(q) \in d(a, q)$, and (2) if $q \sim_a q'$ then $s_a(q) = s_a(q')$. A collective strategy is uniform if it contains only uniform individual strategies. Again, the function $out(q, S_A)$ returns the set of all paths that may result from agents A executing collective strategy S_A from state q. The semantics of ATL$_{ir}$ formulae can be defined as follows:

- $M, q \models \langle\langle A \rangle\rangle_{ir} \bigcirc \varphi$ iff there is a uniform collective strategy S_A such that, for every $a \in A$, q' such that $q \sim_a q'$, and $\lambda \in out(S_A, q')$, we have $M, \lambda[1] \models \varphi$;
- $M, q \models \langle\langle A \rangle\rangle_{ir} \Box \varphi$ iff there exists S_A such that, for every $a \in A$, q' such that $q \sim_a q'$, and $\lambda \in out(S_A, q')$, we have $M, \lambda[i]$ for every $i \geq 0$;
- $M, q \models \langle\langle A \rangle\rangle_{ir} \varphi \mathcal{U} \psi$ iff there exist S_A such that, for every $a \in A$, q' such that $q \sim_a q'$, and $\lambda \in out(S_A, q')$, there is $i \geq 0$ for which $M, \lambda[i] \models \psi$, and $M, \lambda[j] \models \varphi$ for every $0 \leq j < i$.

That is, $\langle\langle A \rangle\rangle_{ir} \varphi$ holds iff A have a uniform collective strategy, such that for every path that can possibly result from execution of the strategy *according to at least one agent from* A, φ is the case.

7.3 Models and Model Checking

In this section, we present and discuss various (existing) representations of systems that can be used for modeling and model checking. We believe that the two most important points of reference are in this case: (1) the modeling formalism (i.e., the logic and the semantics we use), and (2) the phenomenon, or more generally, the domain we are going to model (to which we will often refer as the "real world"). Our aim is a representation which is reasonably close to the real world (i.e., it is sufficiently compact and grounded), and still not too far away from the formalism (so that it e.g. easily allows for theoretical analysis of computational problems). We begin with discussing the merits of "explicit" models – in our case, these are transition systems, concurrent game structures and CEGS s, presented in the previous section.

7.3.1 Explicit Models

Obviously, an advantage of explicit models is that they are very close to the semantics of our logics (simply because they *are* the semantics). On the other hand, they are in many ways difficult to use to describe an actual system:

- Exponential size: temporal models usually have an exponential number of states with respect to any higher-level description (e.g. Boolean variables, n-ary attributes etc.). Also, their size is exponential in the number of processes (or agents) if the evolution of a system results from joint (synchronous or asynchronous) actions of several active entities [92]. For CGS s the situation is even worse: here, also the num-

ber of transitions is exponential, even if we fix the number of states.[3] In practice, this means that such representations are very seldom scalable.

- Explicit models include no modularity. States in a model refer to global states of the system; transitions in the model correspond to global transitions as well, i.e., they represent (in an atomic way) *everything* that may happen in one single step, regardless of who has done it, to whom, and in what way.

- Logics like ATL are often advertised as frameworks for modeling and reasoning about open computational systems. Ideally, one would like the elements of such a system to have as little interdependencies as possible, so that they can be "plugged" in and out without much hassle, for instance when we want to test various designs or implementations of the active component. In the case of a multi-agent system the need is perhaps even more obvious. We do not only need to "re-plug" various designs of a single agent in the overall architecture; we usually also need to change (e.g., increase) the number of agents acting in a given environment without necessarily changing the design of the whole system. Unfortunately, ATL models are anything but open in this sense.

Theoretical complexity results for explicit models are as follows. Model checking CTL and CTLK is P-complete, and can be done in time $O(ml)$, where m is the number of transitions in the model, and l is the length of the formula [30]. Alternatively, it can be done in time $O(n^2l)$, where n is the number of states. Model checking ATL is P-complete wrt. m, l and $\mathbf{\Delta_3^P}$-complete wrt. n, k, l (k being the number of agents) [8, 78, 95]. Model checking ATL$_{ir}$ is $\mathbf{\Delta_2^P}$-complete wrt. m, l and $\mathbf{\Delta_3^P}$-complete wrt. n, k, l [121, 82].

7.3.2 Compressed Representations

Explicit representation of all states and transitions is inefficient in many ways. An alternative is to represent the state/transition space in a symbolic way [102, 103].

Such models offer some hope for feasible model checking properties of open/multi-agent systems, although it is well known that they are compact only in a fraction of all cases.[4] For us, however, they are insufficient for another reason: they are merely *optimized representations of explicit models*. Thus, they are neither more open nor better grounded: they were meant to optimize implementation rather than facilitate design or modeling methodology.

[3] Another class of ATL models, alternating transition systems [7] represent transitions in a more succinct way. While we still have exponentially many states in an ATS, the number of transitions is simply quadratic wrt. to states (like for CTL models). Unfortunately, ATS are even less modular and harder to design than concurrent game structures, and they cannot be easily extended to handle incomplete information (cf. [52]).

[4] Representation R of an explicit model M is *compact* if the size of R is logarithmic with respect to the size of M.

7.3.3 Interpreted Systems

Interpreted systems [58, 42] are held by many as a prime example of *computationally grounded* models of distributed systems. An interpreted system can be defined as a tuple $IS = \langle St_1, ..., St_k, St_{env}, \mathcal{R}, \pi \rangle$. $St_1, ..., St_k$ are local state spaces of agents $1, ..., k$, and St_{env} is the set of states of the environment. The set of global states is defined as $St = St_1 \times ... \times St_k \times St_{env}$; $\mathcal{R} \subseteq St \times St$ is a transition relation, and $\pi : \Pi \rightarrow 2^{St}$. While the transition relation encapsulates the (possible) evolution of the system over time, the epistemic dimension is defined by the local components of each global state: $\langle q_1, ..., q_k, q_{env} \rangle \sim_i \langle q'_1, ..., q'_k, q_{env} \rangle$ iff $q_i = q'_i$.

It is easy to see that such a representation is modular and compact as far as we are concerned with *states*. Moreover, it gives a natural ("grounded") approach to knowledge, and suggests an intuitive methodology for modeling epistemic states. Unfortunately, the way transitions are represented in interpreted systems is neither compact, nor modular, nor grounded: the temporal aspect of the system is given by a joint transition function, exactly like in explicit models. This is not without a reason: if we separate activities of the agents too much, we cannot model *interaction* in the framework any more, and interaction is the most interesting thing here. But the bottom line is that the temporal dimension of an interpreted system has exponential representation. And it is almost as difficult to "plug" components in and out of an interpreted system, as for an ordinary CTL or ATL model, since the "local" activity of an agent is completely merged with his interaction with the rest of the system.

7.3.4 Concurrent Programs

The idea of *concurrent programs* has been long known in the literature on distributed systems. Here, we use the formulation from [92]. A concurrent program P is composed of k concurrent processes, each described by a labeled transition system $P_i = \langle St_i, Act_i, \mathcal{R}_i, \Pi_i, \pi_i \rangle$, where St_i is the set of local states of process i, Act_i is the set of local actions, $\mathcal{R}_i \subseteq St_i \times Act_i \times St_i$ is a transition relation, and Π_i, π_i are the set of local propositions and their valuation. The behavior of program P is given by the product automaton of $P_1, ..., P_k$ under the assumption that processes work asynchronously, actions are interleaved, and synchronization is obtained through common action names.

Concurrent programs have several advantages. First of all, they are modular and compact. They allow for "local" modeling of components – much more so than interpreted systems (not only states, but also actions are local here). Moreover, they allow for representing explicit interaction between local transitions of reactive processes, like willful communication, and synchronization. On the other hand, they do not allow for representing implicit, "incidental", or not entirely benevolent interaction between processes. For example, if we want to represent the act of pushing somebody, the pushed object must explicitly execute an action of "being pushed", which seems somewhat ridiculous. Side effects of actions are also not easy to model. Still, this is a minor complaint in the context of CTL, because for temporal logics we are only interested in the flow of *transitions*, and not in the underlying *actions*. For temporal reasoning about k asynchronous processes with no

implicit interaction, concurrent programs seem just about perfect.

The situation is different when we talk about autonomous, pro-active components (like agents), acting together (cooperatively or adversely) in a common environment – and we want to address their strategies and abilities. Now, particular actions are no less important than the resulting transitions. Actions may influence other agents' local states without their consent, they may have side effects on other agents' states etc. Passing messages and/or calling procedures is by no means the only way of interaction between agents. Moreover, the availability of actions (to an agent) should not depend on the actions that *will* be executed by other agents at the same time – these are the *outcome states* that may depend on these actions! Finally, we would often like to assume that agents act synchronously. In particular, all agents play simultaneously in concurrent game structures. But, assuming synchrony and autonomy of actions, *synchronization* can no longer be a means of coordination.

To sum up, we need a representation which is very much like concurrent programs, but allows for modeling agents that play synchronously, and which enables modeling more sophisticated interaction between agents' actions. The first postulate is easy to satisfy, as we show in the following section. The second will be addressed in Section 7.4.

We note that model checking CTL against concurrent programs is PSPACE-complete in the number of local states and the length of the formula [92].

7.3.5 Synchronous CP and Simple Reactive Modules

The semantics of ATL is based on synchronous models where availability of actions does not depend on the actions currently executed by the other players. A slightly different variant of concurrent programs can be defined via synchronous product of programs, so that all agents play simultaneously.[5] Unfortunately, under such interpretation, no direct interaction between agents' actions can be modeled at all.

Definition 51 *A* synchronous concurrent program *consists of k concurrent processes $P_i = \langle St_i, Act_i, \mathcal{R}_i, \Pi_i, \pi_i\rangle$ with the following unfolding to a CGS:* $\mathrm{Agt} = \{1, ..., k\}$, $St = \prod_{i=1}^{k} St_i$, $Act = \bigcup_{i=1}^{k} Act_i$, $d(i, \langle q_1, ..., q_k\rangle) = \{\alpha_i \mid \langle q_i, \alpha_i, q_i'\rangle \in \mathcal{R}_i$ *for some* $q_i' \in St_i\}$, $o(\langle q_1, ..., q_k\rangle, \alpha_1, ..., \alpha_k) = \langle q_1', ..., q_k'\rangle$ *such that* $\langle q_i, \alpha_i, q_i'\rangle \in \mathcal{R}_i$ *for every* i; $\Pi = \bigcup_{i=1}^{k} \Pi_i$, *and* $\pi(p) = \pi_i(p)$ *for* $p \in \Pi_i$.

We note that the *simple reactive modules* (SRML) from [131] can be seen as a particular implementation of synchronous concurrent programs.

Definition 52 *A SRML system is a tuple $\langle \Sigma, \Pi, m_1, \ldots, m_k\rangle$, where $\Sigma = \{1, \ldots, k\}$ is a set of* modules *(or* agents*), Π is a set of Boolean variables, and, for each $i \in \Sigma$, we have $m_i = \langle ctr_i, init_i, update_i\rangle$, where $ctr_i \subseteq \Pi$. Sets $init_i$ and $update_i$ consist of* guarded commands *of the form $\phi \rightsquigarrow v_1' := \psi_1; \ldots; v_k' := \psi_k$, where every $v_j \in ctr_i$, and $\phi, \psi_1, \ldots, \psi_k$ are propositional formulae over Π. It is required that $ctr_1, \ldots ctr_k$ partitions Π.*

[5] The concept is not new, of course, and has already existed in folk knowledge, although we failed to find an explicit definition in the literature.

The idea is that agent i controls the variables ctr_i. The $init$ guarded commands are used to initialize the controlled variables, while the $update$ guarded commands can change their values in each round. A guarded command is *enabled* if the guard ϕ is true in the current state of the system. In each round an enabled update guarded command is executed: each ψ_j is evaluated against the current state of the system, and its logical value is assigned to v_j. Several guarded commands being enabled at the same time model non-deterministic choice. Model checking ATL for SRML has been proved EXPTIME-complete in the size of the model and the length of the formula [131].

7.3.6 Concurrent Epistemic Programs

Concurrent programs (both asynchronous and synchronous) can be used to encode epistemic relations too – exactly in the same way as interpreted systems do [118]. That is, when unfolding a concurrent program to a model of CTLK or ATL$_{ir}$, we define that $\langle q_1, ..., q_k \rangle \sim_i \langle q'_1, ..., q'_k \rangle$ iff $q_i = q'_i$. Model checking CTLK for concurrent epistemic programs is PSPACE-complete [118]. SRML can be also interpreted in the same way; then, we would assume that every agent can see only the variables he controls.

Concurrent epistemic programs are modular and have a "grounded" semantics. They are usually compact (albeit not always: for example, an agent with perfect information will always blow up the size of such a program). Still, they inherit all the problems of concurrent programs with perfect information, discussed in Section 7.3.4: limited interaction between components, availability of local actions depending on the actual transition etc. The problems were already important for agents with perfect information, but they become even more crucial when agents have only limited knowledge of the current situation. One of the most important applications of logics that combine strategic and epistemic properties is verification of communication protocols (e.g., in the context of security). Now, we may want to, e.g., check agents' ability to pass an information between them, without letting anybody else intercept the message. The point is that the *action* of intercepting is by definition enabled; we just look for a protocol in which the *transition* of "successful interception" is never carried out. So, availability of actions *must* be independent of the actions chosen by the other agents under incomplete information. On the other hand, interaction is arguably the most interesting feature of multi-agent systems, and it is really hard to imagine models of strategic-epistemic logics, in which it is not possible to represent communication.

7.3.7 Reactive Modules

Reactive modules [5] can be seen as a refinement of concurrent epistemic programs (primarily used by the MOCHA model checker [9]), but they are much more powerful, expressive and grounded. We have already mentioned a very limited variant of RML (i.e., SRML). The vocabulary of RML is very close to implementations (in terms of general computational systems): the *modules* are essentially collections of variables, states are just valuations of variables; events/actions are variable updates. However, the sets of variables

controlled by different agents can overlap, they can change over time etc. Moreover, reactive modules support incomplete information (through *observability* of variables), although it is not the main focus of RML. Again, the relationship between sets of observable variables (and to sets of controlled variables) is mostly left up to the designer of a system. Agents can act synchronously as well as asynchronously.

To sum up, RML define a powerful framework for modeling distributed systems with various kinds of synchrony and asynchrony. However, we believe that there is still a need for a simpler and slightly more abstract class of representations. First, the framework of RML is technically complicated, involving a number auxiliary concepts and their definitions. Second, it is not always convenient to represent *all* that is going on in a multi-agent system as reading and/or writing from/to program variables. This view of a multi-agent system is arguably close to its computer implementation, but usually rather distant from the real world domain – hence the need for a more abstract, and more conceptually flexible framework. Third, the separation of the "local" complexity, and the complexity of interaction is not straightforward. Our new proposal, more in the spirit of interpreted systems, takes these observations as the starting point. The proposed framework is presented in Section 7.4.

7.4 Modular Interpreted Systems

The idea behind distributed systems (multi-agent systems even more so) is that we deal with several *loosely* coupled components, where most of the processing goes on *inside* components (i.e., locally), and only a small fraction of the processing occurs *between* the components. Interaction is crucial (which makes concurrent programs an insufficient modeling tool), but it usually consumes much less of the agent's resources than local computations (which makes the explicit transition tables of CGS, CEGS, and interpreted systems an overkill). Modular interpreted systems, proposed here, extrapolate the modeling idea behind interpreted systems in a way that allows for a tight control of the interaction complexity.

Definition 53 *A* modular interpreted system *(MIS) is defined as a tuple*

$$S = \langle \mathbb{A}\mathrm{gt}, env, Act, \mathcal{I}n \rangle,$$

where $\mathbb{A}\mathrm{gt} = \{a_1, ..., a_k\}$ *is a set of agents,* env *is the environment,* Act *is a set of actions, and* $\mathcal{I}n$ *is a set of symbols called* interaction alphabet. *Each agent has the following internal structure:*

$$a_i = \langle St_i, d_i, out_i, in_i, o_i, \Pi_i, \pi_i \rangle, \text{ where:}$$

- St_i *is a set of local states,*
- $d_i : St_i \rightarrow 2^{Act}$ *defines local availability of actions; for convenience of the notation, we additionally define the set of* situated actions *as* $D_i = \{\langle q_i, \alpha \rangle \mid q_i \in St_i, \alpha \in d_i(q_i)\}$,

- out_i, in_i *are* interaction functions*;* $out_i : D_i \to \mathcal{I}n$ *refers to the influence that a given situated action (of agent* a_i*) may possibly have on the external world, and* $in_i : St_i \times \mathcal{I}n^k \to \mathcal{I}n$ *translates external manifestations of the other agents (and the environment) into the "impression" that they make on* a_i*'s transition function depending on the local state of* a_i*,*
- $o_i : D_i \times \mathcal{I}n \to St_i$ *is a (deterministic) local transition function,*
- Π_i *is a set of local propositions of agent* a_i *where we require that* Π_i *and* Π_j *are disjunct when* $i \neq j$*, and*
- $\pi_i : \Pi_i \to 2^{St_i}$ *is a valuation of these propositions.*

The environment $env = \langle St_{env}, out_{env}, in_{env}, o_{env}, \Pi_{env}, \pi_{env}\rangle$ *has the same structure as an agent except that it does not perform actions, and that thus* $out_{env} : St_{env} \to \mathcal{I}n$ *and* $o_{env} : St_{env} \times \mathcal{I}n \to St_{env}$*.*

Within our framework, we assume that every *action* is executed by an *actor*, that is, an agent. As a consequence, every actor is explicitly represented in a MIS as an agent, just like in the case of CGS and CEGS. The environment, on the other hand, represents the (passive) context of agents' actions. In practice, it serves to capture the aspects of the global state that are not observable by any of the agents.

The input functions in_i seem to be the fragile spots here: when given explicitly as tables, they have size exponential wrt. the number of agents (and linear wrt. the size of $\mathcal{I}n$). However, we can use, e.g., a construction similar to the one from [95] to represent interaction functions more compactly.

Definition 54 Implicit input function *for state* $q \in St_i$ *is given by a sequence* $\langle\langle\varphi_1, \eta_1\rangle, ..., \langle\varphi_n, \eta_n\rangle\rangle$*, where each* $\eta_j \in \mathcal{I}n$ *is an interaction symbol, and each* φ_j *is a boolean combination of propositions* $\hat{\eta}^i$*, with* $\eta \in \mathcal{I}n$*;* $\hat{\eta}^i$ *stands for "*η *is the symbol currently generated by agent* i*". The input function is now defined as follows:* $in_i(q, \epsilon_1, ..., \epsilon_k, \epsilon_{env}) = \eta_j$ *iff* j *is the lowest index such that* $\{\hat{\epsilon}^1_1, ..., \hat{\epsilon}^k_k, \hat{\epsilon}^{env}_{env}\} \models \varphi_j$*. It is required that* $\varphi_n \equiv \top$*, so that the mapping is effective.*

Remark 33 *Every* in_i *can be encoded as an implicit input function, with each* φ_j *being of polynomial size with respect to the number of interaction symbols (cf. [95]).*

Note that, for some domains, the MIS representation of a system requires exponentially many symbols in the interaction alphabet $\mathcal{I}n$. In such a case, the problem is inherent to the domain, and in_i will have size exponential wrt the number of agents.

7.4.1 Representing Agent Systems with MIS

Let $St_g = (\prod_{i=1}^k St_i) \times St_{env}$ be the set of all possible global states generated by a modular interpreted system S.

Definition 55 *The unfolding of a MIS* S *for initial states* $Q \subseteq St_g$ *to a CEGS* $cegs(S, Q) = \langle \mathbb{A}\mathrm{gt}', St', \Pi', \pi', Act', d', o', \sim'_1, ..., \sim'_k\rangle$ *is defined as follows:*

- $\mathbb{A}\mathrm{gt}' = \{1, ..., k\}$ *and* $Act' = Act$*,*

- *St' is the set of global states from St_g which are reachable from some state in Q via the transition relation defined by o' (below),*
- $\Pi' = \bigcup_{i=1}^{k} \Pi_i \cup \Pi_{env}$*,*
- *For each $q = \langle q_1, \ldots, q_k, q_{env}\rangle \in St'$ and $i = 1, ..., k, env$, we define $q \in \pi'(p)$ iff $p \in \Pi_i$ and $q_i \in \pi_i(p)$,*
- $d'(i, q) = d_i(q_i)$ *for global state $q = \langle q_1, ..., q_k, q_{env}\rangle$,*
- *The transition function is constructed as follows. Let $q = \langle q_1, ..., q_k, q_{env}\rangle \in St'$, and $\alpha = \langle \alpha_1, ..., \alpha_k\rangle$ be an action profile s.t. $\alpha_i \in d'(i, q)$. We define $input_i(q, \alpha) = in_i\big(q_i, out_1(q_1, \alpha_1), \ldots, out_{i-1}(q_{i-1}, \alpha_{i-1}), out_{i+1}(q_{i+1}, \alpha_{i+1}), \ldots, out_k(q_k, \alpha_k), out_{env}(q_{env})\big)$ for each agent $i = 1, \ldots, k$, and $input_{env}(q, \alpha) = in_{env}\big(q_{env}, out_1(q_1, \alpha_1), \ldots, out_k(q_k, \alpha_k)\big)$.*
  Then, $o'(q, \alpha) = \langle o_1(\langle q_1, \alpha_1\rangle, input_1(q, \alpha)), \ldots, o_k(\langle q_k, \alpha_k\rangle, input_k(q, \alpha)), o_{env}(q_{env}, input_{env}(q, \alpha))\rangle$;
- *For each $i = 1, ..., k$: $\langle q_1, ..., q_k, q_{env}\rangle \sim'_i \langle q'_1, ..., q'_k, q'_{env}\rangle$ iff $q_i = q'_i$.*[6]

Remark 34 *Note that MIS s can be used as representations of CGS s too. In that case, epistemic relations $\sim'_i$ are simply omitted in the unfolding. We denote the unfolding of a MIS S for initial states Q into a CGS by $cgs(S, Q)$.*

Propositions 82 and 83 state that modular interpreted systems can be used as representations for explicit models of multi-agent systems. On the other hand, these representations are not always compact, as demonstrated by Propositions 84 and 85.

Proposition 82 *For every CEGS M, there is a MIS S^M and a set of global states Q of S^M such that $cegs(S^M, Q)$ is isomorphic to M.*[7]

Proof Let $M = \langle\{1, \ldots, k\}, St, Act, d, o, \Pi, \pi, \sim_1, \ldots, \sim_k\rangle$ be a concurrent epistemic game structure. We construct a MIS $S^M = \langle\{a_1, \ldots, a_k\}, env, Act, \mathcal{I}n\rangle$ with agents $a_i = \langle St_i, d_i, out_i, in_i, o_i, \Pi_i, \pi_i\rangle$ and environment $env = \langle St_{env}, out_{env}, in_{env}, o_{env}, \Pi_{env}, \pi_{env}\rangle$, plus a set $Q \subseteq St_g$ of global states, as follows.

- $\mathcal{I}n = Act \cup St \cup (Act^{k-1} \times St)$,
- $St_i = \{[q]_{\sim_i} \mid q \in St\}$ for $1 \leq i \leq k$ (i.e., St_i is the set of i's indistinguishability classes in M),
- $St_{env} = St$,
- $d_i([q]_{\sim_i}) = d(i, q)$ for $1 \leq i \leq k$ (this is well-defined since $d(i, q) = d(i, q')$ whenever $q \sim_i q'$),
- $out_i([q]_{\sim_i}, \alpha_i) = \alpha_i$ for $1 \leq i \leq k$; $out_{env}(q) = q$,
- $in_i([q]_{\sim_i}, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k, q_{env}) = \langle\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k, q_{env}\rangle$ for $i \in \{1, \ldots, k\}$; $in_{env}(q, \alpha_1 \ldots, \alpha_k) = \langle\alpha_1, \ldots, \alpha_k\rangle$;
  $in_i(\vec{x})$ and $in_{env}(\vec{x})$ are arbitrary for other arguments $\vec{x}$,

[6] This shows another difference between the environment and the agents: the environment does not possess knowledge.

[7] We say that two CEGS are isomorphic if they only differ in the names of states and/or actions.

- $o_i(\langle [q]_{\sim_i}, \alpha_i\rangle, \langle \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k, q_{env}\rangle) = [o(q_{env}, \alpha_1, \ldots, \alpha_k)]_{\sim_i}$ for $1 \leq i \leq k$ and $\alpha_i \in d_i([q]_{\sim_i})$;
  $o_{env}(q, \langle \alpha_1, \ldots, \alpha_k\rangle) = o(q, \alpha_1, \ldots, \alpha_k)$;
  o_i and o_{env} are arbitrary for other arguments,
- $\Pi_i = \emptyset$ for $1 \leq i \leq k$, and $\Pi_{env} = \Pi$,
- $\pi_{env}(p) = \pi(p)$
- $Q = \{\langle [q]_{\sim_1}, \ldots, [q]_{\sim_k}, q\rangle : q \in St\}$

Let $M' = cegs(S^M, Q) = \langle \mathbb{A}\mathrm{gt}', St', Act', d', o', \Pi', \pi', \sim'_1, \ldots, \sim'_k\rangle$. We argue that M and M' are isomorphic by establishing a one-to-one correspondence between the respective sets of states, and showing that the other parts of the structures agree on corresponding states.

First we show that, for any $\hat{q}' = \langle [q']_{\sim_1}, \ldots, [q']_{\sim_k}, q'\rangle \in Q$ and any $\alpha = \langle \alpha_1, \ldots, \alpha_k\rangle$ such that $\alpha_i \in d'(i, \hat{q}')$, we have

$$o'(\hat{q}', \alpha) = \langle [q]_{\sim_1}, \ldots, [q]_{\sim_k}, q\rangle \text{ where } q = o(q', \alpha) \tag{7.1}$$

Let $\hat{q} = o'(\hat{q}', \alpha)$. Now, for any i: $input_i(\hat{q}', \alpha) = in_i([q']_{\sim_i}, out_1([q']_{\sim_1}, \alpha_1), ..., out_{i-1}([q']_{\sim_{i-1}}, \alpha_{i-1}), out_{i+1}([q']_{\sim_{i+1}}, \alpha_{i+1}), \ldots, out_k([q']_{\sim_k}, \alpha_k), out_{env}(q')) = in_i([q']_{\sim_i}, \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k, q') = \langle \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k, q'\rangle$. Similarly, we get that $input_{env}(\hat{q}', \alpha) = \langle \alpha_1, \ldots, \alpha_k\rangle$. Thus we get that $o'(\hat{q}', \alpha) = \langle o_1(\langle [q']_{\sim_1}, \alpha_1\rangle, input_1(\hat{q}', \alpha)), \ldots, o_k(\langle [q']_{\sim_k}, \alpha_k\rangle, input_k(\hat{q}', \alpha)), o_{env}(q', input_{env}(\hat{q}', \alpha))\rangle = \langle [o(q', \alpha_1, \ldots, \alpha_k)]_{\sim_1}, \ldots, [o(q', \alpha_1, \ldots, \alpha_k)]_{\sim_k}, o(q', \alpha_1, \ldots, \alpha_k)\rangle$. Thus, $\hat{q} = \langle [q]_{\sim_1}, \ldots, [q]_{\sim_k}, q\rangle$ for $q = o(q', \alpha_1, \ldots, \alpha_k)$, which completes the proof of (7.1).

We now argue that $St' = Q$. Clearly, $Q \subseteq St'$. Let $\hat{q} \in St'$; we must show that $\hat{q} \in Q$. The argument is on induction on the length of the least o' path from Q to $\hat{q}$. The base case, $\hat{q} \in Q$, is immediate. For the inductive step, $\hat{q} = o'(\hat{q}', \alpha)$ for some $\hat{q}' \in Q$, and then we have that $\hat{q} \in Q$ by (7.1). Thus, $St' = Q$.

Now we have a one-to-one correspondence between St and St': $r \in St$ corresponds to $\langle [r]_{\sim_1}, \ldots, [r]_{\sim_k}, r\rangle \in St'$. It remains to be shown that the other parts of the structures M and M' agree on corresponding states:

- $\mathbb{A}\mathrm{gt}' = \mathbb{A}\mathrm{gt}$,
- $Act' = Act$,
- $\Pi' = \bigcup_{i=1}^{k} \Pi_i \cup \Pi_{env} = \Pi$,
- For $p \in \Pi' = \Pi$: $\langle [q']_{\sim_1}, \ldots, [q']_{\sim_k}, q'\rangle \in \pi'(p)$ iff $q' \in \pi_{env}(p)$ iff $q' \in \pi(p)$ (same valuations at corresponding states),
- $d'(i, \langle [q']_{\sim_1}, \ldots, [q']_{\sim_k}, q'\rangle) = d_i([q']_{\sim_i}) = d(i, q)$,
- It follows immediately from (7.1) and the fact that $Q = St'$ that $o'(\langle [q']_{\sim_1}, \ldots, [q']_{\sim_k}, q'\rangle, \alpha) = \langle [r']_{\sim_1}, \ldots, [r']_{\sim_k}, r'\rangle$ iff $o(q', \alpha) = r'$ (transitions on the same joint action in corresponding states lead to corresponding states),

- $\langle [q']_{\sim_1}, \ldots, [q']_{\sim_k}, q' \rangle \sim'_i \langle [r']_{\sim_1}, \ldots, [r']_{\sim_k}, r' \rangle$ iff $[q']_{\sim_i} = [r']_{\sim_i}$ iff $q' \sim_i r'$ (the accessibility relations relate corresponding states), which completes the proof.

■

Corollary 9 *For every CEGS M, there is an ATL $_{ir}$-equivalent MIS S with initial states Q, that is, for every state q in M there is a state q' in $cegs(S, Q)$ satisfying exactly the same ATL $_{ir}$ formulae, and vice versa.*

Proposition 83 *For every CGS M, there is a MIS S^M and a set of global states Q of S^M such that $cgs(S^M, Q)$ is isomorphic to M.*

Proof Let $M = \langle \mathbb{A}\mathrm{gt}, St, Act, d, o, \Pi, \pi \rangle$ be given. Now, let $\hat{M} = \langle \mathbb{A}\mathrm{gt}, St, Act, d, o, \Pi, \pi, \sim_1, \ldots, \sim_k \rangle$ for some arbitrary accessibility relations $\sim_i$ over St. By Proposition 82, there exists a MIS $S^{\hat{M}}$ with global states Q such that $\hat{M}' = cegs(S^{\hat{M}}, Q)$ is isomorphic to $\hat{M}$. Let M' be the CGS obtained by removing the accessibility relations from $\hat{M}'$. Clearly, M' is isomorphic to M. ■

Corollary 10 *For every CGS M, there is an ATL-equivalent MIS S with initial states Q. That is, for every state q in M there is a state q' in $cgs(S, Q)$ satisfying exactly the same ATL formulae, and vice versa.*

Proposition 84 *The local state spaces in a MIS are not always compact with respect to the underlying concurrent epistemic game structure.*

Proof Take a CEGS M in which agent i has always perfect information about the current global state of the system. When constructing a modular interpreted system S such that $M = cegs(S, Q)$, we have that St_i must be isomorphic with St. ■

The above property is a part of the interpreted systems heritage. The next proposition stems from the fact that explicit models (and interpreted systems) allow for intensive interaction between agents.

Proposition 85 *The size of $\mathcal{I}n$ in S is, in general, exponential with respect to the number of local states and local actions. This is the case even when epistemic relations are not relevant (i.e., when S is taken as a representation of an ordinary CGS).*

Proof Consider a CGS M with agents $\mathbb{A}\mathrm{gt} = \{1, ..., k\}$, global states $St = \prod_{i=1}^{k} \{q_0^i, ..., q_i^i\}$, and actions $Act = \{0, 1\}$, all enabled everywhere. The transition function is defined as $o(\langle q_{j_1}^1, ..., q_{j_k}^k \rangle, \alpha_1, ..., \alpha_k) = \langle q_{l_1}^1, ..., q_{l_k}^k \rangle$, where $l_i = (j_i + \alpha_1 + ... + \alpha_k) \mod i$. Note that M can be represented as a modular interpreted system with succinct local state spaces $St_i = \{q_0^i, ..., q_i^i\}$. Still, the current actions of all agents are relevant to determine the resulting local transition of agent i. ■

We will call items $\mathcal{I}n, out_i, in_i$ the *interaction layer* of a modular interpreted system S; the other elements of S constitute the *local layer* of the MIS. In

this paper we are ultimately interested in model checking complexity with respect to the size of the local layer. To this end, we will assume that the size of interaction layer is polynomial in the number of local states and actions. Note that, by Propositions 84 and 85, not every explicit model submits to compact representation with a MIS. Still, as we declared at the beginning of Section 7.4, we are mainly interested in a modeling framework for systems of loosely coupled components, where interaction is essential, but most processing is done locally anyway. More importantly, the framework of MIS allows for separating the interaction of agents from their local structure to a larger extent. Moreover, we can control and measure the complexity of each layer in a finer way than before. First, we can try to abstract from the complexity of a layer (e.g. like in this paper, by assuming that the other layer is kept within certain complexity bounds). Second, we can also measure separately the *interaction complexity of different agents*.

7.4.2 Modular Interpreted Systems vs. Simple Reactive Modules

In this section we show that simple reactive modules are (as we already suggested) a specific (and somewhat limited) implementation of modular interpreted systems. First, we define our (quite strong) notion of equivalence of representations.

Definition 56 *Two representations are equivalent if they unfold to isomorphic concurrent epistemic game structures. They are CGS-equivalent if they unfold to the same CGS.*

Proposition 86 *For any SRML there is a CGS-equivalent MIS.*

Proof Consider an SRML R with k modules and n variables. We construct $S = \langle \mathbb{A}\mathrm{gt}, Act, \mathcal{I}n\rangle$ with $\mathbb{A}\mathrm{gt} = \{a_1, ..., a_k\}$, $Act = \{\top_1, ..., \top_n, \bot_1, ..., \bot_n\}$, and $\mathcal{I}n = \bigcup_{i=1}^{k} St_i \times St_i$ (the local state spaces St_i will be defined in a moment). Let us assume without loss of generality that $ctr_i = \{x_1, ..., x_r\}$. Also, we consider all guarded commands of i to be of type $\gamma^{\top}_{i,\psi} : \psi \rightsquigarrow x_i := \top$, or $\gamma^{\bot}_{i,\psi} : \psi \rightsquigarrow x_i := \bot$. Now, agent a_i in S has the following components: $St_i = 2^{ctr_i}$ (i.e., local states of a_i are valuations of variables controlled by i); $d_i(q_i) = \{\top_1, ..., \top_r, \bot_1, ..., \bot_r\}$; $out_i(q_i, \alpha) = \langle q_i, q_i\rangle$; $in_i(q_i, \langle q_1, q_1\rangle, ..., \langle q_{i-1}, q_{i-1}\rangle, \langle q_{i+1}, q_{i+1}\rangle, ..., \langle q_k, q_k\rangle) = \langle \{x_i \in ctr_i \mid \langle q_1, ..., q_k\rangle \models \bigvee_{\gamma^{\top}_{i,\psi}} \psi\}, \{x_i \in ctr_i \mid \langle q_1, ..., q_k\rangle \models \bigvee_{\gamma^{\bot}_{i,\psi}} \psi\}\rangle$. To define local transitions, we consider three cases. If $t = f = \varnothing$ (no update is enabled), then $o_i(q_i, \alpha, \langle t, f\rangle) = q_i$ for every action α. If $t \neq \varnothing$, we take any arbitrary $\hat{x} \in t$, and define $o_i(q_i, \top_j, \langle t, f\rangle) = q_i \cup \{x_j\}$ if $x_j \in t$, and $q_i \cup \{\hat{x}\}$ otherwise; $o_i(q_i, \bot_j, \langle t, f\rangle) = q_i \setminus \{x_j\}$ if $x_j \in f$, and $q_i \cup \{\hat{x}\}$ otherwise. Moreover, if $t = \varnothing \neq f$, we take any arbitrary $\hat{x} \in f$, and define $o_i(q_i, \top_j, \langle t, f\rangle) = q_i \cup \{x_j\}$ if $x_j \in t$, and $q_i \setminus \{\hat{x}\}$ otherwise; $o_i(q_i, \bot_j, \langle t, f\rangle) = q_i \setminus \{x_j\}$ if $x_j \in f$, and $q_i \setminus \{\hat{x}\}$ otherwise. Finally, $\Pi_i = ctr_i$, and $q_i \in \pi_i(x_j)$ iff $x_j \in q_i$. ■

The above construction shows that SRML have more compact representation of states than MIS: r_i local variables of agent i give rise to 2^{r_i} local

states. In a way, reactive modules (both simple and "full") are two-level representations: first, the system is represented as a product of modules; next, each module can be seen as a product of its variables (together with their update operations). Note, however, that specification of updates with respect to a single variable in an SRML may require guarded commands of total length $O(2^{\sum_{i=1}^{k} r_i})$. Thus, the representation of transitions in SRML is (in the worst case) no more compact than in MIS, despite the two-level structure of SRML. We observe finally that MIS are more general, because in SRML the current actions of other agents have no influence on the outcome of agent i's current action (although the outcome can be influenced by other agents' current local states).

7.4.3 Model Checking Modular Interpreted Systems

One of our main aims was to study the complexity of symbolic model checking ATL$_{ir}$ in a meaningful way. Following the reviewers' remarks, we state our complexity results only as conjectures. Preliminary proofs can be found in [74].

Conjecture 3 *Model checking ATL for modular interpreted systems is EXPTIME-complete.*

Conjecture 4 *Model checking ATL$_{ir}$ for the class of modular interpreted systems is PSPACE-complete.*

A summary of complexity results for model checking temporal and strategic logics (with and without epistemic component) is given in the table below. The table presents completeness results for various models and settings of input parameters. Symbols n, k, m stand for the number of states, agents and transitions in an explicit model; l is the length of the formula, and n_{local} is the number of local states in a concurrent program or modular interpreted system. The new results, conjectured in this paper, are printed in italics. Note that the result for model checking ATL against modular interpreted systems is an extension of the result from [131].

	m, l	n, k, l	n_{local}, k, l
CTL	P [30]	P [30]	PSPACE [92]
CTLK	P [30, 45]	P [30, 45]	PSPACE [118]
ATL	P [8]	$\mathbf{\Delta_3^P}$ [78, 95]	*EXPTIME*
ATL$_{ir}$	$\mathbf{\Delta_2^P}$ [121, 82]	$\mathbf{\Delta_3^P}$ [82]	*PSPACE*

If we are right, then the results for ATL and ATL$_{ir}$ form an intriguing pattern. When we compare model checking agents with perfect vs. imperfect information, the first problem appears to be much easier against explicit models measured with the number of transitions; next, we get the same complexity class against explicit models measured with the number of states and agents; finally, model checking imperfect information turns out to be *easier* than model checking perfect information for modular interpreted systems. Why can it be so?

First, a MIS unfolds into CEGS and CGS in a different way. In the first case, the MIS is assumed to encode the epistemic relations explicitly (which

makes it explode when we model agents with perfect, or almost perfect information). In the latter case, the epistemic aspect is ignored, which gives some extra room for encoding the transition relation more efficiently. Another crucial factor is the number of available strategies (relative to the size of input parameters). The number of all strategies is exponential in the number of global states; for uniform strategies, there are usually much less of them but still exponentially many in general. Thus, the fact that perfect information strategies can be synthesized incrementally has a substantial impact on the complexity of the problem. However, measured in terms of *local states and agents*, the number of all strategies is *doubly exponential*, while there are "only" exponentially many uniform strategies – which settles the results in favor of imperfect information.

7.5 Conclusions

We have presented a new class of representations for open multi-agent systems. Our representations, called modular interpreted systems, are: *modular*, in the sense that components can be changed, replaced, removed or added, with as little changes to the whole representation as possible; more *compact* than traditional explicit representations; and *grounded*, in the sense that the correspondences between the primitives of the model and the entities being modeled are more immediate, giving a methodology for designing and implementing systems. We also conjecture that the complexity of model checking strategic ability for our representations is higher if we assume perfect information than if we assume imperfect information.

The solutions, proposed in this paper, are not necessarily perfect (for example, the "impression" functions in_i seem to be the main source of non-modularity in MIS, and can be perhaps improved), but we believe them to be a step in the right direction. We also do not mean to claim that our representations should replace more elaborate modeling languages like Promela or reactive modules. We only suggest that there is a need for compact, modular and reasonably grounded models that are more expressive than concurrent (epistemic) programs, and still allow for easier theoretical analysis than reactive modules. We also suggest that MIS might be better suited for modeling simple multi-agent domains, especially for human-oriented (as opposed to computer-oriented) design.

Acknowledgments

We thank the anonymous reviewers and Andrzej Tarlecki for their helpful remarks. Thomas Ågotnes' work on this paper was supported by grants 166525/V30 and 176853/S10 from the Research Council of Norway.

Part IV

Strategic Reasoning for Stochastic Multi-Agent Systems

Chapter 8

A Temporal Logic for Markov Chains

Abstract. Most models of agents and multi-agent systems include information about possible states of the system (that defines relations between states and their external characteristics), and information about relationships between states. *Qualitative* models of this kind assign no numerical measures to these relationships. At the same time, *quantitative models* assume that the relationships are measurable, and provide numerical information about the degrees of relations. In this paper, we explore the analogies between some qualitative and quantitative models of agents/processes, especially those between transition systems and Markovian models.
Typical analysis of Markovian models of processes refers only to the expected utility that can be obtained by the process. On the other hand, modal logic offers a systematic method of describing phenomena by combining various modal operators. Here, we try to exploit linguistic features, offered by propositional modal logic, for analysis of Markov chains and Markov decision processes. To this end, we propose *Markov temporal logic* – a multi-valued logic that extends the branching time logic CTL*.

`Keywords:` Temporal logic, Markov chains, Markov decision processes

8.1 Introduction

There are many different models of agents and multi-agent systems; however, most of them follow a similar pattern. First of all, they include information about possible situations (states of the system) that defines relations between states and their external characteristics (essentially, "facts of life" that are true in these states). Second, they provide information about relationships between states (e.g, possible transitions between states).

Models that share this structure can be, roughly speaking, divided into two classes. *Qualitative models* provide no numerical measures for these relationships. They are widely used as basic models of computational systems,

in semantics of programming languages (including agent-oriented languages), and in specification and verification of systems. Qualitative models seem especially suited for domains in which quantitative information cannot be reliably obtained nor assumed. They are also used to model situations in which the goal of an agent (or the whole system) is not to maximize a measurable output, but rather to achieve a state that matches certain characteristics (specified e.g. by a logical formula).

Quantitative models assume that relationships are measurable, and provide numerical information about the degrees of relations. For the relations between states, the degrees are usually given as probabilities. For the "qualities" of particular states, one often talks about *rewards* or *utilities*. Quantitative representations are used in stochastic modeling (Markov chains), decision theory and reinforcement learning (Markov decision processes), game theory (strategic and extensive game forms) etc. In this paper, we explore analogies between transition systems and Markovian models in order to provide a more expressive language for reasoning about, and specification of agents in stochastic environments.

Analysis of quantitative process models is usually based on the notion of expected reward. Still, other features of Markov chains and Markov decision processes can be also interesting. We propose to use the methodology of propositional modal logic in order to study quantitative properties of systems and processes. *Markov temporal logic* for Markov chains, introduced in Section 8.4, is our first step in this direction. We also briefly consider two extensions of the logic: for Markov decision processes (where a single decision maker is present) and for multi-agent Markov decision processes (in which many agents can play simultaneously).

8.1.1 Related Work

Related work includes research on multi-valued logics, especially fuzzy logics [145, 54], probabilistic logics [110, 119], and multi-valued modal logics [50, 36]. Of the latter, [90] is particularly relevant, as it defines a multi-valued version of CTL*, with propositions and accessibility relations taking values from a finite quasi-Boolean algebra. Still, the approach of [90] is too abstract to give an account of quantitative analysis of processes (e.g., by operators that compute the expected and/or average truth value along a given path).

Logics of probability [56] are also related to the phenomena we study here. Important examples of such logics are two probabilistic variants of CTL: PCTL [59] for real time, and pCTL* [10] for discrete time; both allow to express probability bounds for a specified behavior. However, logics of probability do not use the machinery of multi-valued logics. More importantly, like probabilistic logics, they focus on the probabilities of events (e.g., behaviors), and it is often hard to attribute an intuitive meaning to combinations (or patterns) of different probability values. In contrast, we will argue in Section 8.2.3 that combining utilities has a very natural commonsense interpretation.

Our work comes very close to [33], where the "Discounted CTL" (DCTL) is proposed. In fact, our Markov temporal logic directly extends the ideas behind DCTL; a more detailed comparison is presented in Section 8.6. The

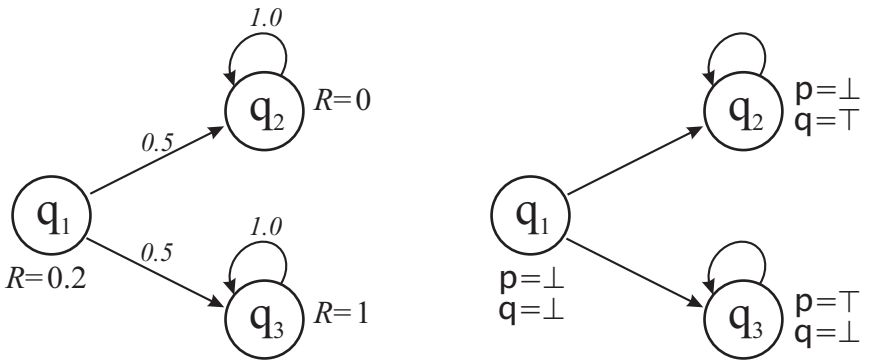


Figure 8.1: (A) Markov chain. (B) Unlabeled transition system

variant of multi-valued CTL from [97], where the domain of truth values can be any c-semiring (rather than simply the interval $[0, 1]$ of real numbers), is also relevant. While it does not address quantitative analysis of processes directly, the choice of c-semirings makes such analysis possible (at least in principle). It may be interesting to consider a similar generalization of our framework in the future.

8.2 Looking for Analogies

We begin with drawing some analogies between the quantitative and qualitative approaches to computational systems. In particular, we are interested in similarities between Markovian models of processes and transition systems.

8.2.1 Quantitative vs. Qualitative Models

The simplest Markovian models are *Markov chains* [100, 88, 53], discrete-time stochastic processes in which the next state of the system depends only on the current state and possibly the current action(s), but it does not directly depend on the past states of the system. A formal definition is given in Section 8.3.2. An example Markov chain is depicted in Figure 8.1, together with an unlabeled transition system. It is easy to see the similarities. First, states in the Markov chain are assigned real reward values R, and states in the transition system are assigned valuations of atomic propositions $\mathsf{p}, \mathsf{q}, \ldots$. Moreover, both kinds of structures include a set of states and a (single) binary transition relation on states; however, in the Markov chain, tuples of the relation are annotated with transition probabilities.

Markov decision processes [16, 15] can be seen as an extension of Markov chains, where several actions are available in each state. We observe that Markov decision processes are very much like labeled transition systems. In both cases, the action-transition structure can be modeled by a number of binary relations on states (one relation per action), although the elements of relations in Markov decision processes are annotated with probability values

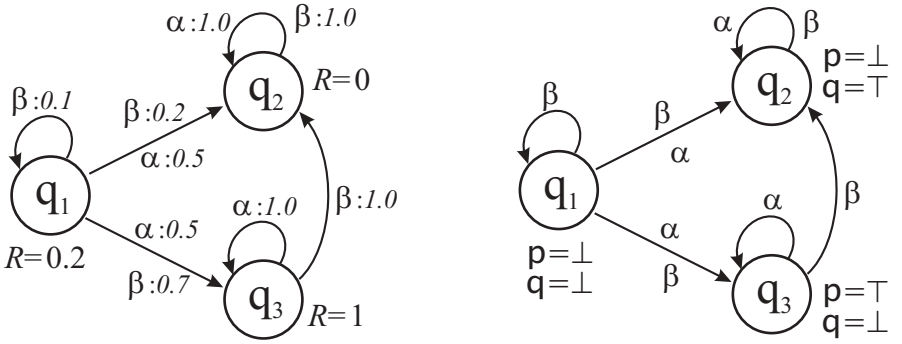


Figure 8.2: (A) Markov decision process. (B) Labeled transition system

(cf. Figure 8.2). We also observe the similarity between multi-agent Markov decision processes from [20] and concurrent game structures from [8].

8.2.2 Quantitative vs. Qualitative Descriptions

The tradition of decision theory and reinforcement learning puts forward the quantitative notion of expected utility which represents the average of "what we can get" for all possible executions of the process. At the same time, logical approaches are usually concerned with "limit properties" like the existence of an execution that displays a specific temporal pattern. Logical frameworks are not very well suited to coping with models that involve probabilities: the existence of a particular kind of execution may be of little interest if this kind of execution is unlikely to happen. It does not mean, however, that these "limit properties" are irrelevant: in some cases we do want to e.g. make sure that there is no path violating an important security property. The point we are trying to make in this paper is that *both* kinds of properties are interesting and worth using to describe processes.

One of the nicer features of temporal logics – especially branching-time logics like CTL and CTL* – is that they offer a systematic approach in which properties of particular paths (executions) are distinguished from the properties of sets of paths (e.g., the set of all executions of a process). The first kind of properties is facilitated by *temporal operators* like "always" ($\Box$), "eventually" ($\Diamond$), "next" ($\bigcirc$) etc. The second kind is based on *path quantifiers* like "for all paths" (A) and "there is a path" (E). Both kinds of operators can be combined: e.g., $\mathsf{E}\Box\mathsf{safe}$ says "there is a path such that the system is always in a safe state". The same approach can be employed within the quantitative framework. For instance, besides the expected value of cumulative future reward, we can ask of the maximal (or minimal) cumulative reward. Or, we might be concerned with the expected value of minimal guaranteed reward etc. We propose a precise semantics for such combinations (and a semantics of interplay between qualitative and quantitative properties) in Section 8.4.

8.2.3 Logical operators as Minimizers and Maximizers

Note that – when truth values represent utility of an agent – temporal operators "sometime" and "always" have a very natural interpretation. "Sometime p" ($\Diamond p$) can be rephrased as "p is achievable in the future". Thus, under the assumption that agents want to obtain as much utility as possible, it is natural to view the operator as maximizing the utility value along a given temporal path. Similarly, "always p" ($\Box p$) can be rephrased as "p is guaranteed from now on". In other words, $\Box p$ asks for the minimal value of p on the path. On a more general level, every universal quantifier is essentially a minimizer of truth values, while existential quantifiers can be seen as maximizers. Thus, Aγ ("for all paths γ") minimizes the utility specified by γ across all paths that can occur, etc. Also, conjunction and disjunction can be seen as a minimizer and a maximizer: $\varphi \vee \psi$ reads easily as "the utility that can be achieved through φ or ψ", while $\varphi \wedge \psi$ reads as "utility guaranteed by both φ and ψ".

Of course, the idea of defining semantics of conjunction and disjunction through functions $\min$ and $\max$, respectively, is not new: the same semantic approach is used e.g. in fuzzy logic [145, 54]. Also, interpreting quantifiers as outcome maximization/minimization operators, can be traced back to the game semantics of classical logic [64, 98].

8.3 Basic Models: Markov Chains and Markov Decision Processes

Markov chains have been proposed to represent and study properties of processes in which transitions can be described in terms of probabilities. Markov chains are often used for generation of semi-random sequences of words, symbols or events (algorithms generating spam messages are a good example here). For these applications, states of a system (chain) play mostly technical role, as we are mainly after the events being generated. However, Markov chains can be also used to model and analyze existing processes (especially as parts of *Markov decision processes*, perhaps the most popular models of reinforcement learning). In that case, we are usually interested in properties of the states: either qualitative (i.e., some facts being true or false in different states of the process) or quantitative (representing utilities or rewards that the process is expected to yield in particular states). Even more importantly, we are interested in how these (qualitative or quantitative) properties accumulate as the system progresses in time.

8.3.1 Domain

A *domain* $D = \langle U, \top, \bot, {}^{-}\rangle$ consists of: (1) a set $U \subseteq \mathbb{R}$ of *utility values* (or simply *utilities*); (2) special values $\top, \bot$ standing for the logical truth and falsity, respectively; $\hat{U} = U \cup \{\top, \bot\}$ will be called the *extended utility set*; and, finally, (3) a complement function ${}^{-} : \hat{U} \to \hat{U}$. A domain should satisfy the following conditions:

1. $U \subseteq \mathbb{R}$;
2. The operations of addition and multiplication have their typical properties on $\hat{U}$, and $\hat{U}$ is closed under averaging, i.e., for every probability distribution P over $\hat{U}$ (discrete or continuous), $\sum_{u \in \hat{U}} u\, P(u) \in \hat{U}$;
3. U is closed under complement: if $u \in U$ then $\overline{u} \in U$;
4. Complement reverts the classical truth values: $\overline{\top} = \bot$ and $\overline{\bot} = \top$;
5. $\top \geq 0$;
6. $\bot \leq u$ and $\top \geq u$ for all $u \in \hat{U}$;[1]
7. The complement is quasi-boolean wrt $\max, \min$, i.e., for every $u_1, u_2, u \in \hat{U}$: $\overline{\max(u_1, u_2)} = \min(\overline{u_1}, \overline{u_2})$, $\overline{\min(u_1, u_2)} = \max(\overline{u_1}, \overline{u_2})$, $\overline{u_1} \leq \overline{u_2}$ iff $u_2 \leq u_1$, and $\overline{\overline{u}} = u$.

In the rest of the paper, we will assume that $U = [0, 1], \top = 1, \bot = 0, \overline{u} = 1 - u$. This closely resembles the setting in [33]. Admittedly, using 0 and 1 to represent "false" and "true" has a long tradition in logic; there is also a tradition of using values between 0 and 1 in multi-valued logics.

8.3.2 Markov Chains

Typically, a Markov chain is a directed graph with probabilistic transition relation. In our definition, we include also a device for assigning states with utilities and/or propositional values. This is done through *utility fluents* which generalize atomic propositions from modal logic, in the sense that they can also take real numbers as their values.

Definition 57 (Markov chain) *A* Markov chain *over domain $D = \langle U, \top, \bot, \overline{\ } \rangle$, and a set of utility fluents Π is a tuple $M = \langle St, \tau, \pi \rangle$, where:*

- *St is a set of* states *(we will assume that the set is finite and nonempty throughout the rest of the paper);*
- *$\tau : St \times St \to [0, 1]$ is a* stochastic transition relation *that assigns each pair of states q_1, q_2 with a probability $\tau(q_1, q_2)$ that, if the system is in q_1, it will change its state to q_2 in the next moment. For every $q_1 \in St$, $\tau(q_1, \cdot)$ is assumed to be a probability distribution, i.e. $\sum_{q \in St} \tau(q_1, q) = 1$.*
  By abuse of notation, we will sometimes write $\tau(q)$ to denote the set of states accessible in one step from q, i.e. $\{q' \mid \tau(q, q') > 0\}$.
- *$\pi : \Pi \times St \to \hat{U}$ is a* valuation of utility fluents.

Example 38 (Gene model) *Consider the following extension of the "gene model" from [53]. A trait in animals of a particular species is governed by a pair of genes, each of whom may be of type G or g. Very often the GG and Gg types are indistinguishable in appearance; we say that type G dominates type g. Thus, an individual can have the* dominant *combination GG,* recessive *combination gg, or* hybrid *combination Gg (which is genetically the same as gG).*

[1] Note that this implies that $\max(u, \top) = \top$, $\min(u, \top) = u$, $\min(u, \bot) = \bot$, and $\max(u, \bot) = u$ for all $u \in \hat{U}$.

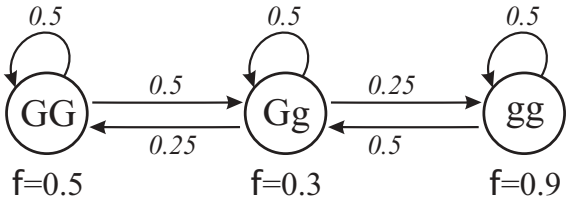


Figure 8.3: Markov chain for the gene model

Mating of two animals produces an offspring that inherits one gene of the pair from each parent, and the basic assumption of genetics is that these genes are selected at random, independently of each other. Suppose that we breed animals by starting with an individual of known genetic character and mate it with a hybrid. We assume that there is at least one offspring. Then, at each round, a random offspring is chosen and mated with a hybrid, and so on. Suppose also that a statistical study of survival produced the following fitness function for individuals of the species (in relation to genotype): $f(GG) = 0.5$*,* $f(Gg) = 0.3$*, and* $f(gg) = 0.9$ *– i.e., the individuals with recessive genes are the fittest, and hybrids are the least fit of all. Furthermore, utility fluent* f *is used to represent fitness values. A Markov chain that models the process is shown in Figure 8.3.*

A *run* in Markov chain M is an infinite sequence of states $q_0 q_1 \ldots$ such that each q_{i+1} can follow q_i with a non-zero probability, i.e., for every $i = 0, 1, \ldots$ we have $\tau(q_i, q_{i+1}) > 0$. We denote the set of all runs in M by $\mathcal{R}_M$. The set of runs starting from state q is denoted by $\mathcal{R}_M(q)$.[2] Let $\lambda = q_0 q_1 ...$ be a run and $i \in \mathbb{N}_0$. Then: $\lambda[i] = q_i$ denotes the ith position in λ; $\lambda[i..j] = q_i \ldots q_j$ denotes the subpath of λ from position i to j; and $\lambda[i..\infty] = q_i q_{i+1} \ldots$ denotes the infinite subpath of λ from position i on.

Finite prefixes of runs are called *histories*. $\mathcal{H}_M = \{h \mid h = \lambda[0..i]$ for some $\lambda \in \mathcal{R}_M, i\}$ denotes the set of all histories in M. $\mathcal{H}_M(q)$ is the set of histories starting from q; $\mathcal{H}^k_M(q)$ restricts the set further to the histories of length k. Note that each history h can be uniquely identified with the set of runs that "complete" it. By a slight abuse of notation, we will also use h to denote the set, and $\mathcal{H}_M(q)$ to denote all such subsets of $\mathcal{R}_M(q)$. Finally, by $\lambda(h)$ we denote an arbitrary infinite continuation of h (e.g., the run from h which is minimal wrt to alphabetical ordering of runs).

8.3.3 Markov Decision Processes

Markov decision processes extend Markov chains with an explicit action structure: transitions are now connected to actions that generate them.

Definition 58 (Markov decision process) *A* Markov decision process *over domain* $D = \langle U, \top, \bot, ^- \rangle$*, and a set of utility fluents* Π *is a tuple* $\mathcal{M} = \langle St, Act, \tau, \pi \rangle$*, where:* St, π *are like in a Markov chain,* Act *is a nonempty finite set of actions, and* $\tau : St \times Act \times St \to [0, 1]$ *is a stochastic transition relation;* $\tau(q_1, \alpha, q_2)$ *defines the probability that, if the system is in* q_1 *and the agent executes* α*, the next state will*

[2] If the model is clear from the context, the subscripts will be omitted.

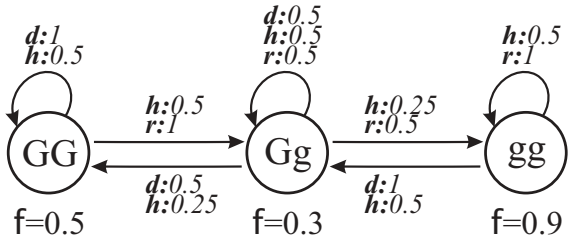


Figure 8.4: Markov decision process that allows for various mating policies

be q_2. For every $q \in St, \alpha \in Act$, we assume that either (1) $\tau(q, \alpha, q') = 0$ for all q' (i.e., α is not enabled in q), or (2) $\tau(q, \alpha, \cdot)$ is a probability distribution.

Additionally, we define $act(q) = \{\alpha \in Act \mid \exists q'.\tau(q, \alpha, q') > 0\}$ as the set of enabled actions in q.

A policy is a conditional plan that specifies future actions of the decision-making agent. Policies can be stochastic as well, thus allowing for randomness in the agent's play.

Definition 59 *A* policy *(or* strategy*) in a Markov decision process $\mathcal{M} = \langle St, Act, \tau, \pi \rangle$ is a function $s : States \times Act \to [0, 1]$ that assigns each state q with a probability distribution over the enabled actions $act(q)$. That is, $s(q, \alpha) \in [0, 1]$ for all $q \in St, \alpha \in act(q)$, and $\sum_{\alpha \in act(q)} s(q, \alpha) = 1$. Values of $s(q, \alpha)$ for $\alpha \notin act(q)$ are irrelevant.*

Policy s is pure *iff for each state q it specifies a single action α (i.e., $s(q, \alpha) = 1$, and $s(q, \alpha') = 0$ for all the other α'). By abuse of notation, we will sometimes write $s(q) = \alpha$ instead of $s(q, \alpha) = 1$ for pure policies.*

The set of all policies in M is denoted by Σ_M. The set of all pure policies in M is denoted by σ_M.

Note that, if the agent's policy is fixed, a Markov decision process reduces to a Markov chain.

Definition 60 *Policy $s : States \times Act \to [0, 1]$* instantiates *MDP $\mathcal{M} = \langle St, Act, \tau, \pi \rangle$ to a Markov chain $\mathcal{M} \dagger s = \langle St', \tau', \pi' \rangle$ with $St' = St$, $\pi' = \pi$, and $\tau'(q, q') = \sum_{\alpha \in act(q)} s(q, \alpha)\, \tau(q, \alpha, q')$.*

Example 39 (Gene model ctd.) *An extension of the "gene model" Markov chain from Example 38 is shown in Figure 8.4. Now, it is possible to mate the offspring with an animal that has dominant genes (action d), recessive genes (action r), or hybrid genes (action h). Note that the pure policy $s(GG) = s(Gg) = s(gg) = h$ instantiates the MDP to the Markov chain from Figure 8.3.*

8.4 MTL$_0$: A Logic of Markov Chains

In this section we present our first take on Markov Temporal Logic (MTL), a logic that allows for flexible reasoning about outcomes of agents acting in stochastic environments. The core of the logic is called MTL$_0$, and addresses

outcomes of Markov chains. Intuitively, MTL$_0$ is a quantitative analogue of the branching-time logic CTL* [38]; we will formalize (and prove) this claim later, in Section 8.4.4.

Operators of MTL$_0$ include path quantifiers $\mathsf{E}, \mathsf{A}, \mathsf{M}$ for the maximal, minimal, and average outcome of a set of temporal paths, respectively, and temporal operators $\Diamond, \Box, \mathsf{m}$ for the maximal, minimal, and average outcome along a given path.[3] Propositional operators follow the same pattern. Besides $\vee, \wedge$ for maximization and minimization of outcomes obtained from different utility channels or related to different goals, we use (after [33]) the "weighted average" operator $\oplus$ which will prove useful when we formulate e.g. fixpoint properties of temporal operators with discount. Additionally, we introduce a "defuzzification" operator $\preccurlyeq$; $\varphi_1 \preccurlyeq \varphi_2$ yields "true" if the outcome of φ_1 is less or equal to φ_2, and "false" otherwise. This provides a neat two-valued interface to the logic. Among other advantages, it allows to define the classical computational problems of validity, satisfiability and model checking for MTL.

8.4.1 Syntax of MTL$_0$

The syntax of MTL$_0$ (parameterized by a set of utility fluents Π) is defined as follows:

$$\begin{aligned} \varphi &::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \oplus_c \varphi \mid \varphi \preccurlyeq \varphi \mid \mathsf{E}\gamma \mid \mathsf{M}\gamma, \\ \gamma &::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \bigcirc_c \gamma \mid \Box_c \gamma \mid \gamma \,\mathcal{U}_c\, \gamma \mid \mathsf{m}_c\gamma. \end{aligned}$$

where $p \in \Pi$ is a utility fluent, and $c \in (0, 1]$ is a discount factor. We will use the symbol $\mathcal{L}state(\Pi)$ to denote the set of "state formulae" φ, and $\mathcal{L}path(\Pi)$ to denote the set of "path formulae" γ.

Additionally, we define the Boolean constants T, F (standing for "true" and "false"), disjunction, and the "sometime" temporal operator $\Diamond$ as below. Except for T, all of them are just standard definitions that can be found in any textbook on temporal logic. We will show in Section 8.4.2 that their semantics corresponds to our intuition also in this setting.

- $\mathrm{T} \equiv p \preccurlyeq p$,
- $\mathrm{F} \equiv \neg\mathrm{T}$,
- $\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2)$,
- $\mathsf{A}\gamma \equiv \neg\mathsf{E}\neg\gamma$,
- $\gamma_1 \vee \gamma_2 \equiv \neg(\neg\gamma_1 \wedge \neg\gamma_2)$,
- $\Diamond_c\gamma \equiv \mathrm{T}\,\mathcal{U}_c\,\gamma$,
- $\varphi_1 \cong \varphi_2 \equiv (\varphi_1 \preccurlyeq \varphi_2) \wedge (\varphi_2 \preccurlyeq \varphi_1)$.

We may also use the following shorthands for discount-free versions of temporal operators: $\bigcirc \equiv \bigcirc_1$, $\Diamond \equiv \Diamond_1$, $\Box \equiv \Box_1$, $\mathcal{U} \equiv \mathcal{U}_1$.

[3] The temporal operators will allow to discount future outcomes with a discount factor c. Also, we will introduce the "until" operator $\mathcal{U}$, which is more general than $\Diamond$.

Example 40 *The following MTL$_0$ formulae define some interesting characteristics of the breeding process from Example 38:* $\mathsf{Mm}_{0.9}\mathsf{f}$ *(expected average fitness with time discount* 0.9*),* $\mathsf{Am}_{0.9}\mathsf{f}$ *(guaranteed average fitness with the same discount factor),* $\mathsf{M}\Box\mathsf{f}$ *(expected minimal undiscounted fitness), and* $\mathsf{A}\Diamond\mathsf{f}$ *(guaranteed maximal fitness).*

8.4.2 Semantics of MTL$_0$

The main idea behind MTL$_0$ is to treat formulae in a sufficiently general way, so that they can represent both quantitative utilities and qualitative truth values referring to something which is *completely* true or false, like a task that has been completely achieved. Besides advantages in terms of modeling, this allows to freely mix qualitative and quantitative properties, which (hopefully) makes the resulting semantics elegant and powerful. Thus, we are going to treat complex formulae as fluents, just like the atomic utility fluents from Π, through a valuation function that assigns formulae with extended utility values from $\hat{U}$.

Let $M = \langle St, \tau, \pi\rangle$ be a Markov chain over domain $D = \langle U, \top, \bot, \overline{\ }\rangle$ and a set of utility fluents Π. The truth value of formulae in M is determined by the valuation function $[\cdot] : (St \times \mathcal{L}state(\Pi)) \cup (\mathcal{R} \times \mathcal{L}path(\Pi)) \to \hat{U}$, defined below. We will omit M in $[\cdot]_{M,q}, [\cdot]_{M,\lambda}$ when the model is clear from the context.

- $[p]_q = \pi(p, q)$, for $p \in \Pi$;
- $[\neg\varphi]_q = \overline{[\varphi]_q}$;
- $[\varphi_1 \wedge \varphi_2]_q = \min([\varphi_1]_q, [\varphi_2]_q)$;
- $[\varphi_1 \oplus_c \varphi_2]_q = (1-c)\cdot[\varphi_1]_q + c \cdot [\varphi_2]_q$;
- $[\varphi_1 \preccurlyeq \varphi_2]_q = \top$ if $[\varphi_1]_q \leq [\varphi_2]_q$ and $\bot$ otherwise;
- $[\varphi]_{M,\lambda} = [\varphi]_{M,\lambda[0]}$.
- $[\neg\gamma]_\lambda = \overline{[\gamma]_\lambda}$;
- $[\gamma_1 \wedge \gamma_2]_\lambda = \min([\gamma_1]_\lambda, [\gamma_2]_\lambda)$;
- $[\bigcirc_c \gamma]_\lambda = c \cdot [\gamma]_{\lambda[1..\infty]}$;
- $[\Box_c\gamma]_{M,\lambda} = \inf_{i=0,1,\ldots}\{c^i[\gamma]_{M,\lambda[i..\infty]}\}$;
- $[\gamma_1 \,\mathcal{U}_c\, \gamma_2]_\lambda =$
  $\sup_{i=0,1,\ldots}\left\{\min(\ \min_{0\leq j<i}\{c^j[\gamma_1]_{\lambda[j..\infty]}\},\ c^i[\gamma_2]_{\lambda[i..\infty]})\right\}$;
- The Markovian temporal operator m_c produces the average discounted reward along the given run:

$$[\mathsf{m}_c\gamma]_\lambda = \begin{cases} (1-c)\sum_{i=0}^{\infty} c^i[\gamma]_{\lambda[i\ldots\infty]} & \text{if } c < 1 \\ \lim_{i\to\infty} \frac{1}{i+1}\sum_{j=0}^{i}[\gamma]_{\lambda[i\ldots\infty]} & \text{if } c = 1 \end{cases}$$

- $[\mathsf{E}\gamma]_q = \sup\{[\gamma]_\lambda \mid \lambda \in \mathcal{R}(q)\}$;
- The Markovian path quantifier $\mathsf{M}\gamma$ produces the expected truth value γ across all the possible runs (from now on). Given M, q, we first define the probability space $\langle \mathcal{R}(q), \mathcal{H}(q), pr\rangle$ induced by the next-state transition probabilities τ (cf. also [33, 88]). In this space, elementary outcomes are runs from $\mathcal{R}(q)$, events are sets of runs that share the same

finite prefix (i.e., ones from $\mathcal{H}(q)$), and the probability measure $pr : \mathcal{H}(q) \to [0,1]$ is defined as $pr(q_0 \dots q_1) = \tau(q_0, q_1) \cdot \dots \cdot \tau(q_{i-1}, q_i)$. Then, we use the valuation of γ as the random variable; the truth value of $\mathsf{M}\gamma$ is defined as its expected value:

$$[\mathsf{M}\gamma]_q = \lim_{k\to\infty} \sum_{h\in\mathcal{H}^k(q)} [\gamma]_{\lambda(h)} \tau(h[0], h[1]) \cdot \dots \cdot \tau(h[k-1], h[k]).$$

Example 41 *The valuations of the MTL_0 formulae from Example 40 for the breeding process from Figure 8.3 are as follows.* $[\mathsf{Mm}_{0.9}\mathsf{f}]_{GG} = 0.484$, $[\mathsf{Mm}_{0.9}\mathsf{f}]_{Gg} = 0.480$, *and* $[\mathsf{Mm}_{0.9}\mathsf{f}]_{gg} = 0.554$*; i.e., the expected average fitness with time discount* 0.9 *is* $0.484, 0.480, 0.554$ *if we start with dominant, hybrid, and recessive genes, respectively. Moreover,* $[\mathsf{Am}_{0.9}\mathsf{f}]_{GG} = 0.32$, $[\mathsf{Am}_{0.9}\mathsf{f}]_{Gg} = 0.3$, *and* $[\mathsf{Am}_{0.9}\mathsf{f}]_{gg} = 0.36$*: the guaranteed average fitness (with discount) is* 0.32, 0.3, *and* 0.36, *respectively. Finally, the expected minimal undiscounted fitness* $[\mathsf{M}\Box\mathsf{f}]_q = 0.3$ *for all states* q*, and the guaranteed maximal fitness* $[\mathsf{A}\Diamond\mathsf{f}]_q = 0.3$ *for all states* q*.*

Proposition 87 *We note that the derived operators have the following semantic characteristics:*[4]

1. $[\mathrm{T}]_{M,q} = \top$ *for every* M, q*;*
2. $[\mathrm{F}]_{M,q} = \bot$ *for every* M, q*;*
3. $[\varphi_1 \vee \varphi_2]_{M,q} = \max([\varphi_1]_{M,q}, [\varphi_2]_{M,q})$*;*
4. $[\gamma_1 \vee \gamma_2]_{M,\lambda} = \max([\gamma_1]_{M,\lambda}, [\gamma_2]_{M,\lambda})$*;*
5. $[\mathsf{A}\gamma]_{M,q} = \inf\{[\gamma]_{M,\lambda} \mid \lambda \in \mathcal{R}(q)\}$*;*
6. $[\Diamond_c\gamma]_{M,\lambda} = \sup_{i=0,1,\dots}\{c^i[\gamma]_{M,\lambda[i..\infty]}\}$*;*
7. $[\varphi_1 \cong \varphi_2]_{M,q} = \top$ *if* $[\varphi_1]_{M,q} = [\varphi_2]_{M,q}$*, and* $\bot$ *otherwise.*

The undiscounted versions of temporal operators "always" and "sometime" have the usual relationship, but it does not transfer to the discounted case. Moreover, discounted "always" is trivial for many domains.

Proposition 88 1. $[\Box\gamma]_{M,\lambda} = [\neg\Diamond\neg\gamma]_{M,\lambda}$*,*

2. $[\Box_c\gamma]_{M,\lambda} = 0$ *if* $c < 1$ *and* $\hat{U} \subseteq \mathbb{R}_+ \cup \{0\}$*.*

8.4.3 Levels of Truth

Since every domain must include a distinguished value for the classical (complete) truth, validity of formulae can be defined in a straightforward way.

Definition 61 (Levels of validity) *Let M be a Markov chain, q a state in M, and φ a formula of MTL_0. Then:*

- φ *is* true in M, q *(written* $M, q \models \varphi$*) iff* $[\varphi]_{M,q} = \top$*.*
- φ *is* valid in M *(written* $M \models \varphi$*) iff it is true in every state of* M*.*

[4] Proofs of propositions (omitted here due to lack of space) can be found in the technical report [69].

- φ *is* valid for Markov chains *(written* $\models \varphi$*) iff it is valid in every Markov chain* M.

Example 42 *Let* M *be the Markov chain from Figure 8.3 with additional utility fluents* $\mathsf{0.3}$*,* $\mathsf{0.32}$ *and* $\mathsf{0.36}$ *such that* $\pi(\mathsf{0.3}, q) = 0.3$*,* $\pi(\mathsf{0.32}, q) = 0.32$*, and* $\pi(\mathsf{0.36}, q) = 0.36$ *for all* $q \in St$*. Then, we have that* $M, GG \models \mathsf{Am}_{0.9}\mathsf{f} \cong \mathsf{0.32}$*,* $M, Gg \models \mathsf{Am}_{0.9}\mathsf{f} \cong \mathsf{0.3}$*, and* $M, gg \models \mathsf{Am}_{0.9}\mathsf{f} \cong \mathsf{0.36}$*. Moreover, the following formula is valid in* M*:* $M \models \mathsf{0.3} \preccurlyeq \mathsf{Am}_{0.9}\mathsf{f} \wedge \mathsf{Am}_{0.9}\mathsf{f} \preccurlyeq \mathsf{0.36}$*.*

Note that T is valid for Markov chains, while F is true in no M, q. Other examples of validities are: $\mathsf{A}\Box\gamma \cong \mathsf{A}\neg\Diamond\neg\gamma$, $\mathsf{E}\Box\gamma \cong \mathsf{E}\neg\Diamond\neg\gamma$ etc. (cf. Proposition 88.1).

Definition 61 enables the traditional view of MTL_0 that identifies "the logic" with the set of valid formulae of that logic. Moreover, it allows to define the typical decision problems for MTL_0 in a natural way:

- Given a formula φ, the *validity problem* asks if $\models \varphi$;
- Given a formula φ, the *satisfiability problem* asks if there are M, q such that $M, q \models \varphi$;
- Given a model M, state q and formula φ, the *model checking problem* asks if $M, q \models \varphi$. Alternatively, the output of model checking can be defined as the value of $[\varphi]_{M,q}$.

An important corollary of Proposition 87.7 is that the notion of equivalence defined by $\cong$ is strong enough to make equivalent formulae interchangeable on all levels of validity.

Corollary 11 *If* $M, q \models \varphi_1 \cong \varphi_2$*, and* ψ' *is obtained from* ψ *through replacing an occurrence of* φ_1 *by* φ_2*, then* $M, q \models \psi$ *iff* $M, q \models \psi'$*.*

8.4.4 Transition Systems as Markov Chains. Correspondence between MTL_0 and CTL*

Markov chains can be seen as generalizations of transition systems, where quantitative information is added via non-classical values of atomic statements and probabilities of transitions. As action labels are absent in Markov chains, these in fact generalize *unlabeled* transition systems (UTS). In this section, we redefine UTS as a proper subclass of Markov chains, in which all the fluents can assume only classical truth values.

Definition 62 *Let* M *be a Markov chain. Formula* φ *is* propositional *in* M *iff it can take only the values of* $\top, \bot$*, i.e.,* $[\varphi]_{M,q} \in \{\top, \bot\}$ *for all* $q \in St$*.*

Propositions have a simple characterization for Markov chains.

Proposition 89 *Let* M *be a Markov chain and* φ *a formula of* MTL_0*. Then* φ *is propositional in* M *iff formula* $(\varphi \cong \mathrm{F}) \vee (\varphi \cong \mathrm{T})$ *is valid in* M*.*

An *unlabeled transition system* can be defined as a Markov chain with only propositional fluents. This way, we obtain the class of models that are used for qualitative branching-time logics, i.e. CTL and CTL*. Of course, when

interpreting formulae of CTL*, one must also ignore the probabilities that are present in Markov chains. The next two propositions show that MTL_0 strictly generalizes CTL*.

Proposition 90 *Let M be a transition system, and φ a formula of CTL*. Then, $M, q \models_{MTL_0} \varphi$ iff $M, q \models_{CTL^*} \varphi$.*

Proposition 91 *There is a transition system M with states q, q' which cannot be distinguished by any CTL* formula, and can be distinguished by a formula of MTL_0.*

8.4.5 State-Based MTL_0

"CTL without star" (or "vanilla CTL") is the most often used variant of computation tree logic, mainly due to the complexity of its model checking problem and the fact that its semantics can be defined entirely in relation to states. "Vanilla" CTL can be seen as a syntactic restriction of CTL*, in which every temporal modality is preceded by exactly one path quantifier. In this section, we consider a similar syntactic restriction on MTL_0; we call it state-based MTL_0.

Definition 63 State-based MTL_0 *(sMTL_0 in short) is given by the following grammar (where $p \in \Pi$ stands for utility fluents, and $c \in (0, 1]$ for discount factors):*

$$\begin{array}{rcl} \varphi & ::= & p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \oplus_c \varphi \mid \varphi \preccurlyeq \varphi \mid \mathsf{E}\gamma \mid \mathsf{A}\gamma \mid \mathsf{M}\gamma, \\ \gamma & ::= & \bigcirc_c \varphi \mid \Box_c \varphi \mid \varphi \mathcal{U}_c \varphi \mid \mathsf{m}_c\varphi. \end{array}$$

Lemma 21 shows that $\mathsf{M}\bigcirc_c \varphi$ implements the discounted expected value of φ in the next moment. Proposition 92 presents fixpoint characterizations for most modalities of sMTL_0. The results from [33] suggest that $\mathsf{M}\Box_c$ and $\mathsf{M}\mathcal{U}_c$ do not have fixpoint characterizations, but this remains to be formally proven.

Lemma 21 *Let φ be a formula of sMTL_0. Then, $[\mathsf{M}\bigcirc_c \varphi]_q = c \sum_{q' \in \tau(q)} [\varphi]_{q'} \tau(q, q')$.*

Proposition 92 *The following formulae of sMTL_0 are valid:*

- $\mathsf{E}\varphi_1 \mathcal{U}_c \varphi_2 \cong \varphi_2 \vee \varphi_1 \wedge \mathsf{E}\bigcirc_c \mathsf{E}\varphi_1 \mathcal{U}_c \varphi_2$;
- $\mathsf{A}\varphi_1 \mathcal{U}_c \varphi_2 \cong \varphi_2 \vee \varphi_1 \wedge \mathsf{A}\bigcirc_c \mathsf{A}\varphi_1 \mathcal{U}_c \varphi_2$;
- $\mathsf{E}\Box_c\varphi \cong \varphi \wedge \mathsf{E}\bigcirc_c \mathsf{E}\Box_c\varphi$;
- $\mathsf{A}\Box_c\varphi \cong \varphi \wedge \mathsf{A}\bigcirc_c \mathsf{A}\Box_c\varphi$;
- $\mathsf{Em}_c\varphi \cong \varphi \oplus_c \mathsf{E}\bigcirc \mathsf{Em}_c\varphi$;
- $\mathsf{Am}_c\varphi \cong \varphi \oplus_c \mathsf{A}\bigcirc \mathsf{Am}_c\varphi$;
- $\mathsf{Mm}_c\varphi \cong \varphi \oplus_c \mathsf{M}\bigcirc \mathsf{Mm}_c\varphi$.

The above characterizations enable computing the truth values of most sMTL_0 formulae by solving sets of simple equations.

Example 43 *The valuations of formula* $\mathsf{Am}_{0.9}\mathsf{f}$ *for states* GG, Gg, gg *of the "gene model" Markov chain can be derived from the following equations:*

$$[\mathsf{Am}_{0.9}\mathsf{f}]_{GG} = 0.1 \cdot 0.5 + 0.9 \min([\mathsf{Am}_{0.9}\mathsf{f}]_{GG}, [\mathsf{Am}_{0.9}\mathsf{f}]_{Gg}),$$
$$[\mathsf{Am}_{0.9}\mathsf{f}]_{Gg} = 0.1 \cdot 0.3 + 0.9 \min([\mathsf{Am}_{0.9}\mathsf{f}]_{GG}, [\mathsf{Am}_{0.9}\mathsf{f}]_{Gg}, [\mathsf{Am}_{0.9}\mathsf{f}]_{gg}),$$
$$[\mathsf{Am}_{0.9}\mathsf{f}]_{gg} = 0.1 \cdot 0.9 + 0.9 \min([\mathsf{Am}_{0.9}\mathsf{f}]_{Gg}, [\mathsf{Am}_{0.9}\mathsf{f}]_{gg}).$$

8.5 MTL$_1$: A Logic of Markov Decision Processes

The main aim of this paper is to offer a systematic study of temporal operators for Markov chains; the study was presented in the previous section. This section briefly shows how MTL$_0$ can be extended to strategic reasoning about Markov decision processes. We propose to use an explicit strategic quantifier $\langle\langle a \rangle\rangle$, similar to the *cooperation modality* from alternating-time temporal logic ATL. The intuitive meaning of $\langle\langle a \rangle\rangle \varphi$ is "the most that the decision maker can make out of φ". Note that there is always only one agent behind an MDP, so putting his name (e.g., "a") inside the operator is superfluous – but it will make the framework easier to extend to the multi-agent case in the future.

8.5.1 Syntax and Semantics of MTL$_1$

The syntax of MTL$_1$ is given by the following grammar:

$$\begin{aligned} \vartheta &::= p \mid \neg\vartheta \mid \vartheta \wedge \vartheta \mid \vartheta \oplus_c \vartheta \mid \vartheta \preccurlyeq \vartheta \mid \langle\langle a \rangle\rangle \varphi, \\ \varphi &::= \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \oplus_c \varphi \mid \mathsf{E}\gamma \mid \mathsf{M}\gamma, \\ \gamma &::= \vartheta \mid \neg\gamma \mid \gamma \wedge \gamma \mid \bigcirc_c \gamma \mid \Box_c \gamma \mid \gamma \,\mathcal{U}_c\, \gamma \mid \mathsf{m}_c \gamma. \end{aligned}$$

Note that a is just a fixed symbol and not a parameter of the strategic operator.

Let $\mathcal{M} = \langle St, Act, \tau, \pi \rangle$ be a Markov decision process over domain $D = \langle U, \top, \bot, \neg \rangle$ and a set of utility fluents Π. The truth value of formulae in M is determined by the valuation function $[\cdot]$ that extends the valuation of MTL$_0$ formulae from Section 8.4.2 as follows:

- $[p]_{\mathcal{M},q} = \pi(p, q)$, for $p \in \Pi$;
- $[\neg\vartheta]_{\mathcal{M},q} = \overline{[\vartheta]_{\mathcal{M},q}}$;
- $[\vartheta_1 \wedge \vartheta_2]_{\mathcal{M},q} = \min([\vartheta_1]_q, [\vartheta_2]_{\mathcal{M},q})$;
- $[\vartheta_1 \oplus_c \vartheta_2]_{\mathcal{M},q} = (1-c) \cdot [\vartheta_1]_{\mathcal{M},q} + c \cdot [\vartheta_2]_{\mathcal{M},q}$;
- $[\vartheta_1 \preccurlyeq \vartheta_2]_{\mathcal{M},q} = \top$ if $[\vartheta_1]_{\mathcal{M},q} \leq [\vartheta_2]_{\mathcal{M},q}$ and $\bot$ otherwise;
- $[\langle\langle a \rangle\rangle \varphi]_{\mathcal{M},q} = \sup\{[\varphi]_{M \dagger s,q} \mid s \in \Sigma_M\}$;
- $[\vartheta]_{\mathcal{M}\dagger s,\lambda} = [\vartheta]_{\mathcal{M},\lambda[0]}$.

We use the same definitions of derived Boolean and temporal operators as in Section 8.4.1. Additionally, we define $\vartheta_1 \cong \vartheta_2 \equiv \vartheta_1 \preccurlyeq \vartheta_2 \wedge \vartheta_2 \preccurlyeq \vartheta_1$, and $[\![a]\!]\varphi \equiv \neg\langle\langle a \rangle\rangle\neg\varphi$. The following proposition shows that $[\![a]\!]\varphi$ implements the outcome of the worst possible policy with respect to φ.

Proposition 93 $[\![a]\!]\varphi]_{\mathcal{M},q} = \inf_{s\in\Sigma_{\mathcal{M}}}\{[\varphi]_{\mathcal{M}\dagger s,q}\}$.

Example 44 *Let $\mathcal{M}$ be the "gene model" MDP from Figure 8.4. Then, we have e.g.* $[\langle\!\langle a\rangle\!\rangle \mathsf{Mm}_{0.9}\mathsf{f}]_{GG} = 0.762$, $[\langle\!\langle a\rangle\!\rangle \mathsf{Mm}_{0.9}\mathsf{f}]_{Gg} = 0.791$, *and* $[\langle\!\langle a\rangle\!\rangle \mathsf{Mm}_{0.9}\mathsf{f}]_{gg} = 0.9$. *Indeed, using only individuals with recessive genes for mating is the best policy when we want to maximize the expected average fitness discounted with* 0.9.

On the other hand, mating with hybrids proves best if we want to minimize *the expected average fitness (with discount* 0.9*) from state GG on; for states Gg and gg, mating with dominant genes gives the worst expectancy:* $[[\![a]\!] \mathsf{Mm}_{0.9}\mathsf{f}]_{GG} = 0.484$, $[[\![a]\!] \mathsf{Mm}_{0.9}\mathsf{f}]_{Gg} = 0.464$, *and* $[[\![a]\!] \mathsf{Mm}_{0.9}\mathsf{f}]_{gg} = 0.507$.

We observe that various levels of satisfaction and validity of MTL$_1$ formulae (and thus also the typical computational problems) can be defined analogously to Section 8.4.3.

The semantic definition od $\langle\!\langle a\rangle\!\rangle$ refers to the set of all stochastic policies Σ, which suggests that looking for the best policy can be quite a complex task. Is it possible to restrict the search to pure policies only? Unfortunately, it turns out that it is not the case in general. However, we conjecture that an analogous property should hold for the "state-based" fragment of MTL$_1$.

Proposition 94 *Let $\vartheta \equiv \langle\!\langle a\rangle\!\rangle\varphi$ be a formula of MTL$_1$. Then, the equation* $[\langle\!\langle a\rangle\!\rangle\varphi]_{\mathcal{M},q} = \sup_{s\in\sigma_{\mathcal{M}}}\{[\varphi]_{\mathcal{M}\dagger s,q}\}$ *does* not *hold. It does not even hold for* labeled transition systems*, i.e., Markov decision processes where all the utility fluents take only classical truth values $\top, \bot$.*

Conjecture 5 *Let $\vartheta \equiv \langle\!\langle a\rangle\!\rangle\varphi$ be a formula of MTL$_1$ in which every temporal operator is immediately preceded by exactly one path quantifier, and every path quantifier is immediately preceded by exactly one strategic operator. Then:* $[\langle\!\langle a\rangle\!\rangle\varphi]_{\mathcal{M},q} = \sup_{s\in\sigma_{\mathcal{M}}}\{[\varphi]_{\mathcal{M}\dagger s,q}\}$.

8.5.2 Beyond MDP: the Multi-Agent Case

In the more general case, a system can include multiple agents/processes, interacting with each other. Here, we only briefly discuss how Markov temporal logic can be extended to handle such interaction.

On the language level, we propose to extend the strategic operator $\langle\!\langle a\rangle\!\rangle$ to a family of operators $\langle\!\langle A\rangle\!\rangle$, parameterized with groups of agents A. Intuitively $\langle\!\langle A\rangle\!\rangle\varphi$ refers to how much agents A can "make out of" φ by following their best joint policy. This would yield a language similar to the alternating-time temporal logic ATL* from [8], albeit with strategic operators separated from path quantifiers.

On the semantic level, multi-agent Markov decision processes [20] can be used as models. The semantics $\langle\!\langle A\rangle\!\rangle\varphi$ should be of course based on the maximal value of φ with respect to A's joint strategies. However, it is not entirely clear how *the other agents'* actions should be fixed in order to instantiate the MMDP to a Markov chain. One option is to assume that the opponents play a strategy that minimizes φ best. This way, operator $\langle\!\langle A\rangle\!\rangle$ would correspond to the maxmin of the two-player game where A is the (collective) maximizer, and the rest of agents fills in the role of the (collective) minimizer. Still, such a semantics would entail a very strong assumption, namely that the opponents of A must also play only *memoryless* strategies.

8.6 Comparison to DCTL

Markov temporal logic (MTL), proposed in this paper, is in many respects similar to the "Discounted CTL" (DCTL) by de Alfaro and colleagues [33]. This section lists some differences between both logics.

1. In DCTL, the set of truth values is $[0, 1]$. We keep the choice more open: it can be any continuous subset of $\mathbb{R} \cup \{-\infty, +\infty\}$.
2. MTL has more general syntax than DCTL: MTL_0 extends CTL* and MTL_1 extends the single-agent fragment of ATL*, while de Alfaro et al.'s DCTL extends only the "vanilla" CTL.
3. E, A are true *path* quantifiers in our framework, in the sense that they refer to "limit properties" of paths. For aggregation of utilities via expected value, we propose a separate path operator M. In contrast, [33] propose a semantics in which both E, A are based on the expected reward. In consequence, neither universal nor existential quantification on paths is expressible in DCTL for models with quantitative transition relations. One peculiar consequence of such approach is that the DCTL's $\mathsf{E}\gamma$ yields the same truth value as $\mathsf{A}\gamma$ for all Markov chains, which is not the case in our framework. Another consequence is that the semantics of path quantifiers in [33] is different for qualitative and quantitative models, which is not the case in our semantics.
4. MTL includes the operator $\preccurlyeq$, which can serve both as a kind of crisp material implication on fuzzy operands, and as a "defuzzification" operator that maps quantitative characteristics to qualitative descriptions.
5. The last feature allows us to define the notions of satisfaction and validity. Thus, standard problems like satisfiability and validity are properly defined in our framework.
6. MTL includes the full "until" operator $\mathcal{U}$, while DCTL includes only "sometime" ($\Diamond$).
7. We propose only the "path semantics" for MTL. We believe it is more appropriate to introduce fixpoint operators rather than to define two different semantics of the same formulae.
8. In contrast to [33], we do not try to capture strategic properties of the decision-making agent with temporal path quantifiers. Instead, we propose to use an explicit strategic quantifier $\langle\langle a \rangle\rangle$.

In essence: we attempt at a more *systematic* exploration of linguistic features that are offered by propositional modal logic for analysis of Markovian models of agents.

8.7 Conclusions

Two kinds of models are used in multi-agent systems to represent and reason about behavior of agents/processes: quantitative and qualitative ones. In this paper, we suggest that both traditions are complementary rather than competitive. In fact, we believe that an integration of both approaches may

bring a really powerful framework for dealing with multi-agent systems. To this, end, we propose *Markov temporal logic* MTL which can be seen as an extension of "Discounted CTL" from [33]. We show that the simplest version of MTL (for Markov chains) strictly extends the branching-time logic CTL*, and we discuss some fixpoint properties for a "state-based" subset of the logic. Finally, we discuss how the basic logic can be extended to address strategic abilities of agents in Markov decision processes, in a way similar to ATL*.

Chapter 9

A Temporal Logic for Stochastic Multi-Agent Systems

Abstract. Typical analysis of Markovian models of processes refers only to the expected utility that can be obtained by the process. On the other hand, modal logic offers a systematic method of characterizing processes by combining various modal operators. A multivalued temporal logic for Markov chains and Markov decision processes has been recently proposed in [71]. Here, we discuss how it can be extended to the multi-agent case. We relate the resulting logic to existing (two-valued) logics of strategic ability, and present fixpoint characterizations for some natural combinations of strategic and temporal operators.

`Keywords`: temporal logic, multi-agent system, Markov decision process.

9.1 Introduction

There are many different models of agents and multi-agent systems; however, most of them follow a similar pattern. First of all, they include information about possible situations (states of the system) that defines relations between states and their external characteristics (essentially, "facts of life" that are true in these states). Second, they provide information about relationships between states (e.g., possible transitions between states). Models that share this structure can be, roughly speaking, divided into two classes. *Qualitative models* provide no numerical measures for these relationships. *Quantitative models* assume that relationships are measurable, and provide numerical information about the degrees of relations. In [71], we explored analogies between transition systems and Markovian models in order to provide a more expressive language for reasoning about, and specification of agents in stochastic environments. In [72], we tentatively extended the framework

to the multi-agent case. Here, we present some formal results on the multi-agent version of the language.

Analysis of quantitative process models is usually based on the notion of expected reward. On the other hand, logical approaches are most often concerned with "limit properties" like the existence of an execution path that displays a specific temporal pattern. We believe that both kinds of properties are interesting and worth using to describe processes. For instance, besides the expected value of cumulative future reward, we can ask of the maximal (or minimal) cumulative reward. Or, we might be concerned with the expected value of minimal guaranteed reward etc. A typical analysis of multi-agent Markov decision processes is even more constrained, as we assume that all the agents in the system cooperate to achieve a common goal (i.e., maximize their common expected cumulative reward). Our extension allows to study the outcomes that can be obtained by *various* groups of agents.

The roots of our proposal can be traced back to multivalued logics on one hand (e.g., fuzzy logics [145] and probabilistic logics [110, 119]), and (crisp) modal logics of probability [59, 10, 56] on the other. A closer inspiration comes from multi-valued modal logics [50, 36, 90, 33, 97]. Of the latter, [90, 33, 97] are particularly relevant, as they define multi-valued versions of temporal logic. Still, the version of Markov Temporal Logic proposed here is (to our best knowledge) the first multivalued logic for reasoning about strategic abilities of agents in stochastic multi-agent systems.

We begin by recalling the basic idea of Markov Temporal Logic (MTL) from [71] (Section 9.2). The remaining sections present the original contribution of the paper: the syntax and semantics of the multi-agent MTL was only presented at a workshop with informal proceedings [72], and the theoretical results (relationship to ATL*, fixpoint properties) are entirely new.

9.2 Markov Temporal Logic

In this section we recall the idea of Markov Temporal Logic (MTL) from [71]. The logic allows for flexible reasoning about outcomes of agents acting in stochastic environments. The core of the logic is called MTL_0, and addresses outcomes of Markov chains. Intuitively, MTL_0 can be seen as a quantitative analogue of the branching-time logic CTL* [38].

9.2.1 Basic Models: Markov Chains

Typically, a Markov chain [100, 88] is a directed graph with probabilistic transition relation. In our definition, we include also a device for assigning states with utilities and/or propositional values. This is done through *utility fluents* which generalize atomic propositions in modal logic in the sense that they can take both numerical and qualitative truth values.

Definition 64 (Domain of truth values) *A* domain $D = \langle U, \top, \bot, \neg \rangle$ *consists of: (1) a set* $U \subseteq \mathbb{R}$ *of* utility values *(or simply* utilities*); (2) special values* $\top, \bot$ *standing for the logical truth and falsity, respectively;* $\hat{U} = U \cup \{\top, \bot\}$ *will be called the* extended utility set*; and, finally, (3) a complement function*

$^- : \hat{U} \to \hat{U}$. *A domain should satisfy the conditions specified in [71], omitted here for lack of space.*

Definition 65 (Markov chain) *A* Markov chain *over domain* $D = \langle U, \top, \bot, ^- \rangle$, *and a set of utility fluents* Π *is a tuple* $M = \langle St, \tau, \pi \rangle$, *where:*

- St *is a set of* states *(we will assume that the set is finite and nonempty throughout the rest of the paper);*
- $\tau : St \times St \to [0, 1]$ *is a* stochastic transition relation *that assigns each pair of states* q_1, q_2 *with a probability* $\tau(q_1, q_2)$ *that, if the system is in* q_1, *it will change its state to* q_2 *in the next moment. For every* $q_1 \in St$, $\tau(q_1, \cdot)$ *is assumed to be a probability distribution, i.e.* $\sum_{q \in St} \tau(q_1, q) = 1$.
  By abuse of notation, we will sometimes write $\tau(q)$ *to denote the set of states accessible in one step from* q, *i.e.* $\{q' \mid \tau(q, q') > 0\}$.
- $\pi : \Pi \times St \to \hat{U}$ *is a* valuation of utility fluents.

A *run* in Markov chain M is an infinite sequence of states $q_0 q_1 \dots$ such that each q_{i+1} can follow q_i with a non-zero probability. The set of runs starting from state q is denoted by $\mathcal{R}_M(q)$.[1] Let $\lambda = q_0 q_1 ...$ be a run and $i \in \mathbb{N}_0$. Then: $\lambda[i] = q_i$ denotes the ith position in λ, and $\lambda[i..\infty] = q_i q_{i+1} \dots$ denotes the infinite subpath of λ from position i on.

9.2.2 Logical Operators as Minimizers and Maximizers

Note that – when truth values represent utility of an agent – temporal operators "sometime" and "always" have a very natural interpretation. "Sometime p" ($\Diamond p$) can be rephrased as "p is achievable in the future". Thus, under the assumption that agents want to obtain as much utility as possible, it is natural to view the operator as maximizing the utility value along a given temporal path. Similarly, "always p" ($\Box p$) can be rephrased as "p is guaranteed from now on". In other words, $\Box p$ asks for the minimal value of p on the path. On a more general level, every universal quantifier is essentially a minimizer of truth values, while existential quantifiers can be seen as maximizers. Thus, $\mathsf{E}\gamma$ ("there is a path such that γ") maximizes the utility specified by γ across all paths that can occur; likewise, $\mathsf{A}\gamma$ ("for all paths γ") minimizes the value of γ across paths. Also, disjunction and conjunction can be seen as a maximizer and a minimizer: $\varphi \vee \psi$ reads easily as "the utility that can be achieved through φ or ψ", while $\varphi \wedge \psi$ reads as "utility guaranteed by both φ and ψ".

9.2.3 MTL$_0$: A Logic of Markov Chains

Operators of MTL$_0$ include path quantifiers $\mathsf{E}, \mathsf{A}, \mathsf{M}$ for the maximal, minimal, and average outcome of a set of temporal paths, respectively, and temporal operators $\Diamond, \Box, \mathsf{m}$ for the maximal, minimal, and average outcome along

[1] If the model is clear from the context, the subscripts will be omitted.

a given path.[2] Propositional operators follow the same pattern: $\vee, \wedge, \oplus$ refer to maximization, minimization, and weighted average of outcomes obtained from different utility channels or related to different goals. Finally, we have the "defuzzification" operator $\preccurlyeq$, which provides a two-valued interface to the logic. $\varphi_1 \preccurlyeq \varphi_2$ yields "true" if the outcome of φ_1 is less or equal to φ_2, and "false" otherwise. Among other advantages, it allows to define the classical computational problems of validity, satisfiability and model checking for MTL.

Let $Bool(\omega) = \neg\omega \mid \omega \wedge \omega \mid \omega \oplus_c \omega \mid \omega \preccurlyeq \omega$ denote quasi-Boolean combinations of formulae of type ω. The syntax of MTL_0 can be defined by the following production rules:

$$\begin{aligned} \varphi &::= p \mid Bool(\varphi) \mid \mathsf{E}\gamma \mid \mathsf{M}\gamma, \\ \gamma &::= \varphi \mid Bool(\gamma) \mid \bigcirc_c \gamma \mid \Box_c \gamma \mid \gamma \,\mathcal{U}_c\, \gamma \mid \mathsf{m}_c\gamma, \end{aligned}$$

where $p \in \Pi$ is a utility fluent, and c is a discount factor such that $0 < c \leq 1$. Additionally, we define $\varphi_1 \cong \varphi_2 \equiv (\varphi_1 \preccurlyeq \varphi_2) \wedge (\varphi_2 \preccurlyeq \varphi_1)$. Boolean constants T, F ("true", "false"), disjunction, and the "sometime" temporal operator $\Diamond$ are defined in the standard way. The following shorthands are used for discount-free versions of temporal operators: $\bigcirc \equiv \bigcirc_1, \Diamond \equiv \Diamond_1, \Box \equiv \Box_1, \mathcal{U} \equiv \mathcal{U}_1$.

Example 45 *Let* r *be a utility fluent that represents the immediate reward at each state. The following* MTL_0 *formulae define some interesting characteristics of a process:* $\mathsf{Mm}_{0.9}\mathsf{r}$ *(expected average reward with time discount* 0.9*),* $\mathsf{Am}_{0.9}\mathsf{r}$ *(guaranteed average reward with the same discount factor),* $\mathsf{M}\Box\mathsf{r}$ *(expected minimal undiscounted reward), and* $\mathsf{A}\Diamond\mathsf{r}$ *(guaranteed maximal reward).*

The main idea behind MTL_0 is that formulae can refer to both quantitative utilities and qualitative truth values. Thus, we treat complex formulae as fluents, just like the atomic utility fluents from Π, through a valuation function that assigns formulae with extended utility values from $\hat{U}$. Let $M = \langle St, \tau, \pi \rangle$ be a Markov chain over domain $D = \langle U, \top, \bot, \overline{\ } \rangle$ and a set of utility fluents Π. The valuation function $[\cdot]$ is defined below.

- $[p]_{M,q} = \pi(p, q)$, for $p \in \Pi$;
- $[\neg\varphi]_{M,q} = \overline{[\varphi]_{M,q}}$;
- $[\varphi_1 \wedge \varphi_2]_{M,q} = \min([\varphi_1]_{M,q}, [\varphi_2]_{M,q})$;
- $[\varphi_1 \oplus_c \varphi_2]_{M,q} = (1-c) \cdot [\varphi_1]_{M,q} + c \cdot [\varphi_2]_{M,q}$;
- $[\varphi_1 \preccurlyeq \varphi_2]_{M,q} = \top$ if $[\varphi_1]_{M,q} \leq [\varphi_2]_{M,q}$ and $\bot$ else;
- $[\mathsf{E}\gamma]_{M,q} = \sup\{[\gamma]_{M,\lambda} \mid \lambda \in \mathcal{R}(q)\}$;
- The Markovian path quantifier $\mathsf{M}\gamma$ produces the expected truth value γ across all the possible runs, cf. [88] for the formal construction;
- $[\varphi]_{M,\lambda} = [\varphi]_{M,\lambda[0]}$;
- $[\neg\gamma]_{M,\lambda}$, $[\gamma_1 \wedge \gamma_2]_{M,\lambda}$, $[\gamma_1 \oplus_c \gamma_2]_{M,\lambda}$, $[\gamma_1 \preccurlyeq \gamma_2]_{M,\lambda}$: analogous to Boolean combinations of "state formulae" φ;

[2] We allow to discount future outcomes with a discount factor c. Also, we introduce the "until" operator $\mathcal{U}$, which is more general than $\Diamond$.

- $[\bigcirc_c \gamma]_{M,\lambda} = c \cdot [\gamma]_{M,\lambda[1..\infty]}$;
- $[\Box_c \gamma]_{M,\lambda} = \inf_{i=0,1,\ldots}\{c^i [\gamma]_{M,\lambda[i..\infty]}\}$;
- $[\gamma_1 \,\mathcal{U}_c\, \gamma_2]_{M,\lambda} = \sup_{i=0,1,\ldots}\left\{\min(\,\min_{0\leq j<i}\{c^j [\gamma_1]_{M,\lambda[j..\infty]}\},\; c^i [\gamma_2]_{M,\lambda[i..\infty]})\right\}$;
- The Markovian temporal operator m_c produces the average discounted reward along the given run:

$$[\mathsf{m}_c \gamma]_{M,\lambda} =$$

$$\begin{cases} (1-c)\sum_{i=0}^{\infty} c^i [\gamma]_{M,\lambda[i\ldots\infty]} & \text{if } c < 1 \\ \frac{\limsup_{i\to\infty} \frac{1}{i+1}\sum_{j=0}^{i} [\gamma]_{M,\lambda[j\ldots\infty]} + \liminf_{i\to\infty} \frac{1}{i+1}\sum_{j=0}^{i} [\gamma]_{M,\lambda[j\ldots\infty]}}{2} & \text{if } c = 1 \end{cases}$$

9.3 Reasoning about Stochastic Multi-Agent Processes

Strategic abilities were already considered in MTL_1, the version of Markov Temporal Logic for reasoning about Markov decision processes [71]. In consequence, MTL_1 can be seen as a quantitative analogue of the *single-agent* fragment of ATL* [8] with memoryless strategies. In the more general case, a system can include multiple agents/processes, interacting with each other. To address their properties, a family of operators $\langle\langle A\rangle\rangle$ can be used, parameterized with groups of agents A. Intuitively, $\langle\langle A\rangle\rangle\varphi$ refers to how much agents A can "make out of" φ by following their best joint policy. This yields a language similar to the alternating-time temporal logic ATL* from [8], albeit with strategic operators separated from path quantifiers.

Markov decision processes [16, 15] extend Markov chains with an explicit action structure: transitions are generated by actions of an (implicit) decision maker. *Multi-agent Markov decision processes* (MMDP) [20] extend Markov decision processes to the multi-agent setting: transitions are now labeled by combinations of agents' actions. We observe the similarity between MMDP's and concurrent game structures which are the models of ATL* (cf. Figure 9.1).

As models for our multi-agent MTL, we will use a refinement of MMDP's similar to the version of Markov chains presented in Section 9.2.1. The semantics of $\langle\langle A\rangle\rangle\varphi$ is based on maximization of the value of φ with respect to A's joint strategies. We assume that the opponents play a strategy that minimizes φ most. This way, operator $\langle\langle A\rangle\rangle$ corresponds to the maxmin of the two-player game where A is the (collective) maximizer, and the rest of agents fills in the role of the (collective) minimizer. Note that such a semantics entails that the opponents of A must also play only *memoryless* (i.e., Markovian) strategies.

9.3.1 MTL_2: Syntax

Let $\mathbb{A}\mathrm{gt}$ be the set of all agents. MTL_2 adds to MTL_0 a family of operators $\langle\langle A\rangle\rangle$, one for each group of agents $A \subseteq \mathbb{A}\mathrm{gt}$. Formally, the syntax of MTL_2 is given

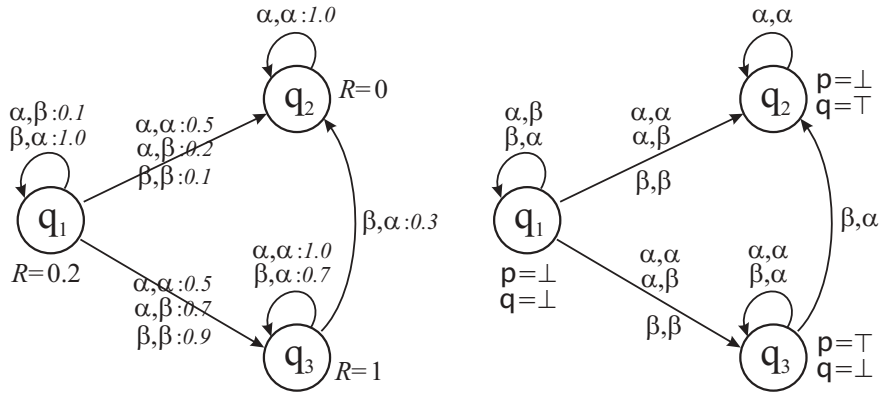


Figure 9.1: (A) Simple MMDP with two agents; (B) Simple concurrent game structure

by the following grammar:

$$\begin{array}{lcl} \vartheta & ::= & p \mid Bool(\vartheta) \mid \langle\!\langle A \rangle\!\rangle \varphi, \\ \varphi & ::= & \vartheta \mid Bool(\varphi) \mid \mathsf{E}\gamma \mid \mathsf{M}\gamma, \\ \gamma & ::= & \varphi \mid Bool(\gamma) \mid \bigcirc_c \gamma \mid \Box_c \gamma \mid \gamma \,\mathcal{U}_c\, \gamma \mid \mathsf{m}_c\gamma. \end{array}$$

An example formula of MTL_2 is $\langle\!\langle 1, 2 \rangle\!\rangle \mathsf{A}\mathsf{m}\mathsf{r}$ which makes agents 1 and 2 maximize the guaranteed average reward r with respect to their available policies.

9.3.2 MTL_2: Semantics

The semantics of MTL_2 is defined for a version of multi-agent Markov decision processes that incorporates qualitative as well as quantitative atomic properties of states.

Definition 66 (MMDP) *A* multi-agent Markov decision process *over domain* $D = \langle U, \top, \bot, ^- \rangle$ *and utility fluents* Π *is a tuple* $\mathcal{M} = \langle \mathbb{A}\text{gt}, St, \{Act_i\}_{i \in \mathbb{A}\text{gt}}, \tau, \pi \rangle$*, where:* St, π *are like in a Markov chain,* $\mathbb{A}\text{gt} = \{1, \ldots, k\}$ *is the set of* agents*,* Act_i *is the set of* individual actions *of agent* i*, and* $Act = \prod_{i \in \mathbb{A}\text{gt}} Act_i$ *is the space of* joint actions *(*action profiles*).* $\tau : St \times Act \times St \to [0, 1]$ *is a stochastic transition relation;* $\tau(q_1, \alpha, q_2)$ *defines the probability that, if the system is in* q_1 *and the agents execute* α*, the next state will be* q_2*. For every* $q \in St, \alpha \in Act$*, we assume that either (1)* $\tau(q, \alpha, q') = 0$ *for all* q' *(i.e.,* α *is not enabled in* q*), or (2)* $\tau(q, \alpha, \cdot)$ *is a probability distribution. Additionally, we define* $act(q) = \{\alpha \in Act \mid \exists q'.\tau(q, \alpha, q') > 0\}$ *as the set of enabled action profiles in* q*.*

For a joint action α, we define α^i to denote agent i's individual part in α, and we extend the notation to sets of joint actions and agents. Also, let $\mathcal{A}$ be a set of action profiles, and α a collective action of agents A. Then, $\mathcal{A}|\alpha = \{\beta \in \mathcal{A} \mid \beta^A = \alpha\}$ is the set of action profiles that include α.

A policy is a conditional plan that specifies future actions of an agent. Policies can be stochastic as well, thus allowing for randomness in the agent's play.

Definition 67 *An* individual strategy *(*policy*) of agent i is a function $s_i : St \times Act_i \to [0,1]$ that assigns each state q with a probability distribution over i's enabled actions $act(q)^i$. That is, $s(q,\alpha_i) \in [0,1]$ for all $q \in St, \alpha_i \in act(q)^i$, and $\sum_{\alpha_i \in act(q)^i} s(q,\alpha_i) = 1$. Values of $s(q,\alpha_i)$ for $\alpha_i \notin act(q)^i$ are irrelevant. The set of all $i's$ strategies is denoted by Σ_i. A* collective strategy *s_A for team $A \subseteq \mathbb{A}\mathrm{gt}$ is simply a tuple of individual strategies, one per agent from A. The set of all A's collective strategies is given by $\Sigma_A = \prod_{i \in A} \Sigma_i$. The set of all* strategy profiles *in a model is given by $\Sigma = \Sigma_{\mathbb{A}\mathrm{gt}}$.*

For a collective strategy s, we define s^i as the i's individual part in s. We also extend the notation to sets of agents.

Definition 68 *Policy $s \in \Sigma_A$* instantiates *multi-agent Markov decision process $\mathcal{M} = \langle \mathbb{A}\mathrm{gt}, St, \{Act_i\}_{i \in \mathbb{A}\mathrm{gt}}, \tau, \pi \rangle$ to a simpler multi-agent MDP $\mathcal{M} \dagger s = \langle \mathbb{A}\mathrm{gt} \setminus A, St, \{Act_i\}_{i \in \mathbb{A}\mathrm{gt} \setminus A}, \tau', \pi \rangle$ with*

$$\tau'(q,\alpha,q') = \sum_{\alpha' \in (act(q)|\alpha)} \left(\prod_{i \in A} s^i(q,\alpha')\right) \tau(q,\alpha',q').$$

If $A = \mathbb{A}\mathrm{gt}$, then s instantiates $\mathcal{M}$ to a Markov chain.

The semantics of MTL_2 extends that of MTL_0 with the following clauses:

- $[p]_{\mathcal{M},q} = \pi(p,q)$, for $p \in \Pi$;
- $[\neg\vartheta]_{\mathcal{M},q}, [\vartheta_1 \wedge \vartheta_2]_{\mathcal{M},q}, [\vartheta_1 \oplus_c \vartheta_2]_{\mathcal{M},q}, [\vartheta_1 \preccurlyeq \vartheta_2]_{\mathcal{M},q}$: analogous as for "state formulae" φ;
- $[\langle\!\langle A \rangle\!\rangle \varphi]_{\mathcal{M},q} = \sup_{s \in \Sigma_A} \inf_{t \in \Sigma_{\mathbb{A}\mathrm{gt} \setminus A}} \{[\varphi]_{\mathcal{M} \dagger \langle s,t \rangle, q}\}$;

In order to keep consistent with qualitative logics of strategic ability, we assume that instantiation of an MMDP by a policy s is "soft" in the sense that nested strategic operators discard previous instantiations and instantiate the original model again: $[\langle\!\langle A \rangle\!\rangle \varphi]_{\mathcal{M} \dagger s,q} = [\langle\!\langle A \rangle\!\rangle \varphi]_{\mathcal{M},q}$.

Example 46 *Consider the multi-agent Markov decision process from Figure 9.1A, consisting of two agents (1 and 2). If the agents cooperate, they can maximize the expected achievable reward quite successfully, as $[\langle\!\langle 1,2 \rangle\!\rangle \mathsf{M}\Diamond R]_{q_1} = 0.9$ (best policy: both agents play β in q_1 with probability 1; the choices at other states are irrelevant). If agent 1 is to maximize the expected achievable reward on his own, against adversary behavior of agent 2, then he is bound to be less successful: $[\langle\!\langle 1 \rangle\!\rangle \mathsf{M}\Diamond R]_{q_1} = 0.6$. Also, in this case agent 1 should employ a different policy, namely play α in q_1 with probability 1.*

9.4 Formal Results

The semantics of MTL, presented in the previous section, portrays it as a language of arithmetic expressions that can be used to define numerical characteristics of Markov processes. However, MTL can be also seen as a *logic*, i.e. a set of *sentences* that are true in some contexts, and false (at least to a degree) in others. This view allows us to use the conceptual apparatus of mathematical logic to study e.g. the expressivity of the language. Also, we can state interesting properties of the domain (multi-agent stochastic processes) through formulae of MTL. To this end, we first define what it means for a formula to be valid and/or satisfiable.

9.4.1 Levels of Truth

Since every domain must include a distinguished value for the classical (complete) truth, validity of formulae can be defined in a straightforward way.

Definition 69 (Levels of validity) *Let $\mathcal{M}$ be a multi-agent Markov decision process, q a state in $\mathcal{M}$, and ϑ a formula of MTL$_2$. Then:*

- *ϑ is* true in *$\mathcal{M}, q$ (written $\mathcal{M}, q \models \vartheta$) iff $[\vartheta]_{\mathcal{M},q} = \top$.*
- *ϑ is* valid in *$\mathcal{M}$ (written $\mathcal{M} \models \vartheta$) iff it is true in every state of $\mathcal{M}$.*
- *ϑ is* valid for multi-agent Markov decision processes *(written $\models \vartheta$) iff it is valid in every MMDP $\mathcal{M}$.*
- *Additionally, for path formulae γ, we can say that γ holds on run λ in MMDP $\mathcal{M}$ (written $\mathcal{M}, \lambda \models \gamma$) iff $[\gamma]_{\mathcal{M},\lambda} = \top$.*

The notion of validity helps to express general properties of stochastic multi-agent systems in a neat logical way. Moreover, Definition 69 allows to define the typical decision problems for MTL$_2$ in a natural way:

- Given a formula ϑ, the *validity problem* asks if $\models \vartheta$;
- Given a formula ϑ, the *satisfiability problem* asks if there are $\mathcal{M}, q$ such that $\mathcal{M}, q \models \vartheta$;
- Given a model $\mathcal{M}$, state q and formula ϑ, the *model checking problem* asks if $\mathcal{M}, q \models \vartheta$.

For example, we can search for a model in which agent a can guarantee the average reward r to be at least 0.6 in the long run by solving the satisfiability problem for formula $0.6 \preccurlyeq \langle\langle a \rangle\rangle \mathsf{Amr}$.

We consider model checking the most important of the three problems, since in the analysis of a stochastic system the domain specification is usually given by a procedural representation (rather than axiomatic theory). Some work on model checking multi-valued temporal logics has been reported in [90, 33]. Perhaps even more importantly, computing approximate "solutions" of MDP's is one of the central issues studied by the Markov community. Integration of the two approaches seems a very promising (and exciting) path for future research.

9.4.2 Concurrent Game Structures as MMDP's. Correspondence between MTL$_2$ and ATL*

Multi-agent Markov decision processes can be seen as generalizations of concurrent game structures [8], in which quantitative information is added through non-classical values of atomic statements and probabilities of transitions. Conversely, concurrent game structures can be seen as a subclass of MMDP's with all fluents assuming only classical truth values.

Definition 70 *Let $\mathcal{M}$ be an MMDP. Formula φ is* propositional *in $\mathcal{M}$ iff it can take only the values of $\top, \bot$, i.e., $[\varphi]_{\mathcal{M},q} \in \{\top, \bot\}$ for all $q \in St$. A* concurrent game structure *is an MMDP with only propositional fluents.*

This way, we obtain the class of models that are used for qualitative alternating-time logics, i.e. ATL and ATL*. Of course, when interpreting formulae of qualitative ATL/ATL*, one must as well ignore the probabilities that are present in Markov decision processes. Note also that the semantics of the original ATL/ATL* uses the "history-based" notion of a strategy (i.e., strategies assign choices to *histories* rather than single states), while our MTL_2 is underpinned by a much weaker notion of *memoryless* (or positional) strategies. This makes the two logics formally incomparable. However, we can show that MTL_2 strictly generalizes the memoryless version of ATL*. The latter was studied in [121] under the acronym of ATL_{Ir}* (ATL with Perfect **I**nformation and imperfect **r**ecall), and we will use the name here.

Proposition 95 *Let $\mathcal{M}$ be a transition system, and φ a formula of ATL_{Ir}*. Moreover, let φ' be the result of replacing every occurrence of $\langle\langle A\rangle\rangle$ with $\langle\langle A\rangle\rangle\mathsf{A}$ in φ for all $A \subseteq \mathbb{A}\text{gt}$. Then, $\mathcal{M}, q \models_{MTL_2} \varphi'$ iff $\mathcal{M}, q \models_{ATL_{\text{IR}^*}} \varphi$.*

Proof sketch Let σ_A denote the set of *deterministic* memoryless strategies of group A.[3] The proof follows by induction on the structure of φ; here, we only sketch the induction step for the most important case, namely $\varphi \equiv \langle\langle A\rangle\rangle_{Ir}\gamma$. We recall from [121] the semantics of $\langle\langle A\rangle\rangle_{Ir}$: let $out_{\mathcal{M}}(q,s)$ be the set of paths in $\mathcal{M}$ that can result from execution of strategy s from state q on; then, $\mathcal{M}, q \models \langle\langle A\rangle\rangle_{Ir}\gamma$ iff there is $s \in \sigma_A$ such that for every $\lambda \in out_{\mathcal{M}}(q,s)$ we have $\mathcal{M}, \lambda \models \gamma$.

"$\text{MTL}_2 \Rightarrow \text{ATL}_{\text{Ir}}$*": Let $\mathcal{M}, q \models_{\text{MTL}_2} \langle\langle A\rangle\rangle\mathsf{A}\gamma$. Then, $[\langle\langle A\rangle\rangle\mathsf{A}\gamma]_{\mathcal{M},q} = \top$, and so $\sup_{s\in\Sigma_A}\inf_{t\in\Sigma_{\mathbb{A}\text{gt}\setminus A}}\inf_{\lambda\in\mathcal{R}_{\mathcal{M}\dagger\langle s,t\rangle}(q)}[\gamma]_{\mathcal{M}\dagger\langle s,t\rangle,\lambda} = \top$; let s^* be a strategy that maximizes the above expression. Note that all the state subformulae of γ will be in fact evaluated in the original MMDP $\mathcal{M}$, so we get that $\inf_{t\in\Sigma_{\mathbb{A}\text{gt}\setminus A}}\inf_{\lambda\in\mathcal{R}_{\mathcal{M}\dagger\langle s^*,t\rangle}(q)}[\gamma]_{\mathcal{M},\lambda} = \top$. Thus, $\forall_{t\in\Sigma_{\mathbb{A}\text{gt}\setminus A}}\forall_{\lambda\in\mathcal{R}_{\mathcal{M}\dagger\langle s^*,t\rangle}(q)}[\gamma]_{\mathcal{M},\lambda} = \top$, and by induction we obtain that $\forall_{t\in\Sigma_{\mathbb{A}\text{gt}\setminus A}}\forall_{\lambda\in\mathcal{R}_{\mathcal{M}\dagger\langle s^*,t\rangle}(q)}\mathcal{M}, \lambda \models_{\text{ATL}_{\text{IR}^*}} \gamma$. Now we observe that if $s \in \Sigma_A$ is a randomized strategy and $\lfloor s\rfloor \in \sigma_A$ is any determinization of s then $\mathcal{R}_{\mathcal{M}\dagger\langle\lfloor s\rfloor,t\rangle}(q) \subseteq \mathcal{R}_{\mathcal{M}\dagger\langle s,t\rangle}(q)$, so also for $\lfloor s^*\rfloor$ we have that $\forall_{t\in\Sigma_{\mathbb{A}\text{gt}\setminus A}}\forall_{\lambda\in\mathcal{R}_{\mathcal{M}\dagger\langle\lfloor s^*\rfloor,t\rangle}(q)}\mathcal{M}, \lambda \models_{\text{ATL}_{\text{IR}^*}} \gamma$. Finally, we take t to be the uniform randomized strategy of $\mathbb{A}\text{gt} \setminus A$ since it does not remove any paths from the model: $\mathcal{R}_{\mathcal{M}\dagger\langle\lfloor s^*\rfloor,uniform\rangle}(q) = out_{\mathcal{M}}(q,\lfloor s^*\rfloor)$. In consequence, $\forall_{\lambda\in out_{\mathcal{M}}(q,\lfloor s^*\rfloor)}\mathcal{M}, \lambda \models_{\text{ATL}_{\text{IR}^*}} \gamma$, which concludes this part of the proof.

"ATL_{Ir}* $\Leftarrow \text{MTL}_2$": Let $\mathcal{M}, q \models_{\text{ATL}_{\text{IR}^*}} \langle\langle A\rangle\rangle_{Ir}\gamma$. Then, $\exists_{s\in\sigma_A}\forall_{\lambda\in out_{\mathcal{M}}(q,s)}\mathcal{M}, \lambda \models_{\text{ATL}_{\text{IR}^*}} \gamma$. We take such s. By induction, $\forall_{\lambda\in out_{\mathcal{M}}(q,s)}\mathcal{M}, \lambda \models_{\text{MTL}_2} \gamma$. Take any $t \in \Sigma_{\mathbb{A}\text{gt}\setminus A}$, then $\mathcal{R}_{\mathcal{M}\dagger\langle s,t\rangle}(q) \subseteq out_{\mathcal{M}}(q,s)$, and hence also $\forall_{\lambda\in\mathcal{R}_{\mathcal{M}\dagger\langle s,t\rangle}(q)}\mathcal{M}, \lambda \models_{\text{MTL}_2} \gamma$. As $\sigma_A \subseteq \Sigma_A$, we finally get that $\exists_{s\in\Sigma_A}\forall_{t\in\Sigma_{\mathbb{A}\text{gt}\setminus A}}\forall_{\lambda\in\mathcal{R}_{\mathcal{M}\dagger\langle s,t\rangle}(q)}\mathcal{M}, \lambda \models_{\text{MTL}_2} \gamma$. In consequence, $\sup_{s\in\Sigma_A}\inf_{t\in\Sigma_{\mathbb{A}\text{gt}\setminus A}}\inf_{\lambda\in\mathcal{R}_{\mathcal{M}\dagger\langle s,t\rangle}(q)}[\gamma]_{\mathcal{M}\dagger\langle s,t\rangle,\lambda} = \top$, which concludes the proof. ∎

Proposition 96 *There is a transition system M with states q, q' which cannot be distinguished by any formula of ATL* nor ATL_{Ir}*, and can be distinguished by a formula of MTL_2.*

Proof Consider the transition system in Figure 9.2, which can be seen as a concurrent game structure with a single agent ($\mathbb{A}\text{gt} = \{1\}$) and a single

[3] Recall that Σ_A is the set of all (possibly randomized) memoryless strategies of A.

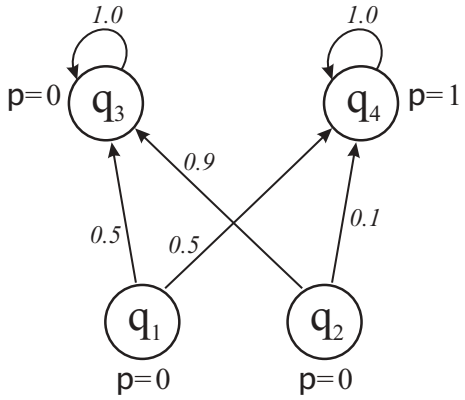


Figure 9.2: MTL_2 vs. ATL*: probabilities matter!

action that can be executed ($Act = \{\alpha\}$). Note that states q_1, q_2 are bisimilar under CTL* bisimulation, so the same CTL* properties hold in both states (cf. e.g. [104]). Since the agent cannot make any real choices, both ATL* and ATL_{Ir}* have no more distinguishing power for this model as CTL*, and hence the same properties of ATL* (resp. ATL_{Ir}*) hold in q_1, q_2 as well.

On the other hand, we have that $[\langle\langle 1\rangle\rangle \mathsf{Mmp}]_{q_1} = 0.5 = [\langle\langle 1\rangle\rangle \mathsf{Em}_{0.5}\mathsf{p}]_{q_1}$, and $[\langle\langle 1\rangle\rangle \mathsf{Mmp}]_{q_2} = 0.1 \neq 0.5 = [\langle\langle 1\rangle\rangle \mathsf{Em}_{0.5}\mathsf{p}]_{q_2}$. Thus, for $\varphi \equiv (\langle\langle 1\rangle\rangle \mathsf{Mmp} \cong \langle\langle 1\rangle\rangle \mathsf{Em}_{0.5}\mathsf{p})$, we have $q_1 \models \varphi$ and $q_2 \not\models \varphi$ (and even $q_2 \models \neg\varphi$). ■

The above example shows that a proper notion of bisimulation for Markov decision processes must take into account transition probabilities.

9.4.3 State-Based Formulae and Bellman Equations

"ATL without star" (or "vanilla ATL") is the most often used variant of alternating-time temporal logic, mainly due to the complexity of its model checking problem and the fact that its semantics can be defined entirely in relation to states. "Vanilla" ATL can be seen as a syntactic restriction of ATL*, in which every temporal modality is preceded by exactly one path quantifier. In this section, we consider a similar syntactic restriction on MTL_2; we call it state-based MTL_2.

Definition 71 State-based MTL_2 *(sMTL_2 in short) is given as follows:*

$$\begin{array}{lcl} \vartheta & ::= & p \mid Bool(\vartheta) \mid \langle\langle A\rangle\rangle\varphi, \\ \varphi & ::= & \mathsf{E}\gamma \mid \mathsf{M}\gamma, \\ \gamma & ::= & \bigcirc_c \vartheta \mid \Box_c\vartheta \mid \vartheta\,\mathcal{U}_c\,\vartheta \mid \mathsf{m}_c\vartheta. \end{array}$$

Proposition 97 presents fixpoint characterizations for most modalities of sMTL_2. Note that the last validity from the list is in fact a modal formulation of *Bellman equation*, which is the basic law used in analysis of Markov decision processes. The other formulae can be seen as variants of the equation for non-standard analysis based on minimal/maximal rather than average rewards. The results from [33] suggest that $\langle\langle A\rangle\rangle \mathsf{M}\Box_c$ and $\langle\langle A\rangle\rangle \mathsf{M}\mathcal{U}_c$ do not have fixpoint characterizations, but this remains to be formally proven.

Proposition 97 *The following formulae of* sMTL_2 *are valid:*

- $\langle\!\langle A\rangle\!\rangle \mathsf{E}\Box_c\varphi \cong \varphi \wedge \langle\!\langle A\rangle\!\rangle \mathsf{E}\bigcirc_c \langle\!\langle A\rangle\!\rangle \mathsf{E}\Box_c\varphi$*;*
- $\langle\!\langle A\rangle\!\rangle \mathsf{A}\Box_c\varphi \cong \varphi \wedge \langle\!\langle A\rangle\!\rangle \mathsf{A}\bigcirc_c \langle\!\langle A\rangle\!\rangle \mathsf{A}\Box_c\varphi$*;*
- $\langle\!\langle A\rangle\!\rangle \mathsf{E}\varphi_1 \,\mathcal{U}_c\, \varphi_2 \cong \varphi_2 \vee \varphi_1 \wedge \langle\!\langle A\rangle\!\rangle \mathsf{E}\bigcirc_c \langle\!\langle A\rangle\!\rangle \mathsf{E}\varphi_1 \,\mathcal{U}_c\, \varphi_2$*;*
- $\langle\!\langle A\rangle\!\rangle \mathsf{A}\varphi_1 \,\mathcal{U}_c\, \varphi_2 \cong \varphi_2 \vee \varphi_1 \wedge \langle\!\langle A\rangle\!\rangle \mathsf{A}\bigcirc_c \langle\!\langle A\rangle\!\rangle \mathsf{A}\varphi_1 \,\mathcal{U}_c\, \varphi_2$*;*
- $\langle\!\langle A\rangle\!\rangle \mathsf{Em}_c\varphi \cong \varphi \oplus_c \langle\!\langle A\rangle\!\rangle \mathsf{E}\bigcirc \langle\!\langle A\rangle\!\rangle \mathsf{Em}_c\varphi$*;*
- $\langle\!\langle A\rangle\!\rangle \mathsf{Am}_c\varphi \cong \varphi \oplus_c \langle\!\langle A\rangle\!\rangle \mathsf{A}\bigcirc \langle\!\langle A\rangle\!\rangle \mathsf{Am}_c\varphi$*;*
- $\langle\!\langle A\rangle\!\rangle \mathsf{Mm}_c\varphi \cong \varphi \oplus_c \langle\!\langle A\rangle\!\rangle \mathsf{M}\bigcirc \langle\!\langle A\rangle\!\rangle \mathsf{Mm}_c\varphi$*.*

Proof sketch We will sketch the proof of the first validity; the others can be proved in an analogous way.

Let $L = [\langle\!\langle A\rangle\!\rangle \mathsf{E}\Box_c\varphi]_{\mathcal{M},q}$ and $R = [\varphi \wedge \langle\!\langle A\rangle\!\rangle \mathsf{E}\bigcirc_c \langle\!\langle A\rangle\!\rangle \mathsf{E}\Box_c\varphi]_{\mathcal{M},q}$. Then:
$R = \min([\varphi]_{\mathcal{M},q},$
$c \cdot \sup_{s\in\Sigma_A} \inf_{t\in\Sigma_{\mathbb{A}\mathrm{gt}\setminus A}} \sup_{q'\in\tau_{\mathcal{M}\dagger\langle s,t\rangle}(q)} \sup_{s'\in\Sigma_A} \inf_{t'\in\Sigma_{\mathbb{A}\mathrm{gt}\setminus A}} \{[\mathsf{E}\Box_c\varphi]_{\mathcal{M}\dagger\langle s',t'\rangle,q'}\}).$
Moreover, by [71, Proposition 8], we get that
$L = \min([\varphi]_{\mathcal{M},q},\ c \cdot \sup_{s\in\Sigma_A} \inf_{t\in\Sigma_{\mathbb{A}\mathrm{gt}\setminus A}} \sup_{q'\in\tau_{\mathcal{M}\dagger\langle s,t\rangle}(q)} \{[\mathsf{E}\Box_c\varphi]_{\mathcal{M}\dagger\langle s,t\rangle,q'}\}).$
Thus, in order to prove $L = R$, it is sufficient to prove that
$\sup_{s\in\Sigma_A} \inf_{t\in\Sigma_{\mathbb{A}\mathrm{gt}\setminus A}} \sup_{q'\in\tau_{\mathcal{M}\dagger\langle s,t\rangle}(q)} \{[\mathsf{E}\Box_c\varphi]_{\mathcal{M}\dagger\langle s,t\rangle,q'}\}$
$= \sup_{s\in\Sigma_A} \inf_{t\in\Sigma_{\mathbb{A}\mathrm{gt}\setminus A}} \sup_{q'\in\tau_{\mathcal{M}\dagger\langle s,t\rangle}(q)} \sup_{s'\in\Sigma_A} \inf_{t'\in\Sigma_{\mathbb{A}\mathrm{gt}\setminus A}} \{[\mathsf{E}\Box_c\varphi]_{\mathcal{M}\dagger\langle s',t'\rangle,q'}\}.$
The difference between the sides of the equation is that in the left hand side optimal strategies s, t are chosen once (at state q), while in the right hand side strategies are re-evaluated after each step. Let $s*$ be a strategy of A that optimizes L, and let us take s and s' in R to be the same as $s*$ in L. We observe that $\inf_{t\in\Sigma_{\mathbb{A}\mathrm{gt}\setminus A}} \sup_{q'\in\tau_{\mathcal{M}\dagger\langle s*,t\rangle}(q)} \{[\mathsf{E}\Box_c\varphi]_{\mathcal{M}\dagger\langle s*,t\rangle,q'}\}$ is indeed equal to $\inf_{t\in\Sigma_{\mathbb{A}\mathrm{gt}\setminus A}} \sup_{q'\in\tau_{\mathcal{M}\dagger\langle s*,t\rangle}(q)} \inf_{t'\in\Sigma_{\mathbb{A}\mathrm{gt}\setminus A}} \{[\mathsf{E}\Box_c\varphi]_{\mathcal{M}\dagger\langle s*,t'\rangle,q'}\}$. Thus, we obtain that A have at least as good options in R as in L, and hence $L \leq R$.

For the other direction, note that s, t in R are only relevant wrt the agents' actions in state q (later s', t' will be used). By unfolding R, we obtain an infinite sequence of collective action profiles $s_n(q_n), t_n(q_n)$ which maximize (over A's actions) and minimize (over $\mathbb{A}\mathrm{gt} \setminus A$'s actions) the value of $\mathsf{E}\Box_c\varphi$ in the next step. Now we observe that, when the system returns to state q, the same strategies s, t will be again optimal for the respective parties since the same expression will be maximized/minimized. Thus, the sequence of action profiles can be combined into a single pair of memoryless strategies $s*, t*$, which maximizes/minimizes $\mathsf{E}\Box_c\varphi$ as good as the original sequence of strategies. In consequence, also $R \leq L$. ∎

9.5 Conclusions

We extend the Markov Temporal Logic MTL from [71] to handle Markovian models with multiple agents acting in parallel. In terms of formal results, we show that the resulting logic strictly embeds $\mathrm{ATL}_{\mathrm{Ir}}$*, i.e., alternating-time temporal logic with memoryless strategies. We also present fixpoint characterizations for some natural combinations of strategic, path, and temporal

operators, that can be seen as analogues of Bellman equation. The characterizations enable computing the truth values of many MTL_2 formulae by solving sets of simple equations.

Part V

Bibliography

Bibliography

[1] T. Ågotnes. A note on syntactic characterization of incomplete information in atel. In *Procedings of the Workshop on Knowledge and Games, Liverpool*, pages 34–42, 2004.

[2] T. Ågotnes, V. Goranko, and W. Jamroga. Alternating-time temporal logics with irrevocable strategies. In D. Samet, editor, *Proceedings of TARK XI*, pages 15–24, 2007.

[3] Thomas Ågotnes. Action and knowledge in alternating-time temporal logic. *Synthese*, 149(2):377–409, 2006. Section on Knowledge, Rationality and Action.

[4] R. Alur, L. de Alfaro, R. Grossu, T.A. Henzinger, M. Kang, C.M. Kirsch, R. Majumdar, F.Y.C. Mang, and B.-Y. Wang. jMocha: A model-checking tool that exploits design structure. In *Proceedings of ICSE*, pages 835–836. IEEE Computer Society Press, 2001.

[5] R. Alur and T. A. Henzinger. Reactive modules. *Formal Methods in System Design*, 15(1):7–48, 1999.

[6] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 100–109. IEEE Computer Society Press, 1997.

[7] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Lecture Notes in Computer Science*, 1536:23–60, 1998.

[8] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.

[9] R. Alur, T.A. Henzinger, F.Y.C. Mang, S. Qadeer, S.K. Rajamani, and S. Tasiran. MOCHA user manual. In *Proceedings of CAV'98*, volume 1427 of *Lecture Notes in Computer Science*, pages 521–525, 1998.

[10] A. Aziz, V. Singhal, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. It usually works: The temporal logic of stochastic systems. In *Proceedings of CAV*, volume 939 of *LNCS*, pages 155–165, 1995.

[11] M. Bacharach. A theory of rational decision in games. *Erkenntnis*, 27:17–55, 1987.

[12] C. Baier, M. Kwiatkowska, and G. Norman. Computing probability bounds for linear time formulas over concurrent probabilistic systems. *Electronic Notes in Theoretical Computer Science*, 21(19), 1999.

[13] J.L. Balcázar, J. Diaz, and J. Gabarró. *Structural Complexity I*. Springer-Verlag, 1988.

[14] A. Baltag. A logic for suspicious players. *Bulletin of Economic Research*, 54(1):1–46, 2002.

[15] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[16] R. Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6:679–684, 1957.

[17] N. Belnap and M. Perloff. Seeing to it that: a canonical form for agentives. *Theoria*, 54:175–199, 1988.

[18] A. L. Blum and M. L. Furst. Fast planning through graph analysis. *Artificial Intelligence*, 90:281–300, 1997.

[19] G. Bonanno. The logic of rational play in games of perfect information. *Economics and Philosophy*, 7:37–65, 1991.

[20] Craig Boutilier. Sequential optimality and coordination in multiagent systems. In *Proceedings of IJCAI*, pages 478–485, 1999.

[21] T. Brihaye, A. Da Costa, F. Laroussinie, and N. Markey. ATL with strategy contexts and bounded memory. Technical Report LSV-08-14, ENS Cachan, 2008.

[22] J. Broersen, A. Herzig, and N. Troquard. Embedding Alternating-time Temporal Logic in Strategic STIT logic of agency. *Journal of Logic and Computation*, 16(5):559–578, 2006.

[23] N. Bulling. Modal logics for games, time, and beliefs. Master thesis, Clausthal University of Technology, 2006.

[24] N. Bulling and W. Jamroga. Agents, beliefs and plausible behaviour in a temporal setting. Technical Report IfI-06-05, Clausthal University of Technology, 2006.

[25] N. Bulling and W. Jamroga. Agents, beliefs and plausible behaviour in a temporal setting. In *Proceedings of AAMAS'07*, pages 570–577, 2007.

[26] N. Bulling and W.Jamroga. A logic for reasoning about rational agents: Yet another attempt. In L. Czaja, editor, *Proceedings of CS&P*, pages 87–99, 2007.

[27] K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. In *CONCUR*, pages 59–73, 2007.

[28] F. Chu and J. Halpern. On the NP-completeness of finding an optimal strategy in games with common payoffs. *International Journal of Game Theory*, 2001.

[29] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Proceedings of Logics of Programs Workshop*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71, 1981.

[30] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[31] V. Conitzer and T. Sandholm. Complexity results about Nash equilibria. Technical Report CMU-CS-02-135, School of Computer Science, Carnegie-Mellon University, 2002.

[32] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of ACM*, 42(4):857–907, 1995.

[33] L. de Alfaro, M. Faella, T.A. Henzinger, R. Majumdar, and M. Stoelinga. Model checking discounted temporal properties. *Theoretical Computer Science*, 345:139–170, 2005.

[34] L. de Alfaro, T.A. Henzinger, and F.Y.C. Mang. The control of synchronous systems. In *Proceedings of CONCUR 2000*, volume 1877 of *Lecture Notes in Computer Science*, pages 458–473, 2000.

[35] L. de Alfaro, T.A. Henzinger, and F.Y.C. Mang. The control of synchronous systems, part II. In *Proceedings of CONCUR 2001*, volume 2154 of *Lecture Notes in Computer Science*, pages 566–580, 2001.

[36] Steve Easterbrook and Marsha Chechik. A framework for multi-valued reasoning over inconsistent viewpoints. In *International Conference on Software Engineering*, pages 411–420, 2001.

[37] T. Eiter. Oral communication. March 2005.

[38] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier Science Publishers, 1990.

[39] E.A. Emerson and J.Y. Halpern. "sometimes" and "not never" revisited: On branching versus linear time temporal logic. In *Proceedings of the Annual ACM Symposium on Principles of Programming Languages*, pages 151–178, 1982.

[40] E.A. Emerson and J.Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, 30(1):1–24, 1985.

[41] E.A. Emerson and J.Y. Halpern. "sometimes" and "not never" revisited: On branching versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178, 1986.

[42] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press: Cambridge, MA, 1995.

[43] R. Fagin and J.Y. Halpern. Reasoning about knowledge and probability. *Journal of ACM*, 41(2):340–367, 1994.

[44] M. Fisher. *Temporal Logics*. Kluwer, 2006.

[45] M. Franceschet, A. Montanari, and M. de Rijke. Model checking for combined logics. In *Proceedings of the 3rd International Conference on Temporal Logic (ICTL)*, 2000.

[46] N. Friedman and J.Y. Halpern. A knowledge-based framework for belief change, Part I: Foundations. In *Proceedings of TARK*, pages 44–64, 1994.

[47] N. Friedman and J.Y. Halpern. A knowledge-based framework for belief change, Part II: Revision and update. In *Proceedings of KR'94*, pages 190–200, 1994.

[48] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman: San Francisco, 1979.

[49] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1989.

[50] P. Godefroid, M. Huth, and R. Jagadeesan. Abstraction-based model checking using modal transition systems. In *Proceedings of CONCUR*, volume 2154 of *LNCS*, pages 426–440, 2001.

[51] V. Goranko. Coalition games and alternating temporal logics. In J. van Benthem, editor, *Proceedings of TARK VIII*, pages 259–272. Morgan Kaufmann, 2001.

[52] V. Goranko and W. Jamroga. Comparing semantics of logics for multiagent systems. *Synthese*, 139(2):241–280, 2004.

[53] C. M. Grinstead and J. L. Snell. *Introduction to Probability*. Amer Mathematical Society, 1997.

[54] P. Hajek. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.

[55] J. Y. Halpern. Reasoning about knowledge: a survey. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *The Handbook of Logic in Artificial Intelligence and Logic Programming, Volume IV*, pages 1–34. Oxford University Press, 1995.

[56] J. Y. Halpern. A logical approach to reasoning about uncertainty: a tutorial. In X. Arrazola, K. Korta, and F. J. Pelletier, editors, *Discourse, Interaction, and Communication*, pages 141–155. Kluwer, 1998.

[57] J.Y. Halpern. Reasoning about knowledge: a survey. In *Handbook of Logic in Artificial Intelligence and Logic Programming. Vol. 4: Epistemic and Temporal Reasoning*, pages 1–34. Oxford University Press, Oxford, 1995.

[58] J.Y. Halpern and R. Fagin. Modelling knowledge and action in distributed systems. *Distributed Computing*, 3(4):159–177, 1989.

[59] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[60] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.

[61] B.P. Harrenstein, W. van der Hoek, J.-J. Meyer, and C. Witteveen. A modal characterization of Nash equilibrium. *Fundamenta Informaticae*, 57(2–4):281–321, 2003.

[62] P. Harrenstein, W. van der Hoek, J-J. Meijer, and C. Witteveen. Subgame-perfect Nash equilibria in dynamic logic. In M. Pauly and A. Baltag, editors, *Proceedings of the ILLC Workshop on Logic and Games*, pages 29–30. University of Amsterdam, 2002. Tech. Report PP-1999-25.

[63] A. Herzig and N. Troquard. Knowing how to play: Uniform choices in logics of agency. In *Proceedings of AAMAS'06*, pages 209–216, 2006.

[64] J. Hintikka. *Logic, Language Games and Information*. Clarendon Press : Oxford, 1973.

[65] M. Huth and M. Z. Kwiatkowska. Quantitative analysis and model checking. In *Logic in Computer Science*, pages 111–122, 1997.

[66] W. Jamroga. Some remarks on alternating temporal epistemic logic. In B. Dunin-Keplicz and R. Verbrugge, editors, *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, pages 133–140, 2003.

[67] W. Jamroga. *Using Multiple Models of Reality. On Agents who Know how to Play Safer*. PhD thesis, University of Twente, 2004.

[68] W. Jamroga. On the relationship between playing rationally and knowing how to play: A logical account. In C. Freksa, M. Kohlhase, and K. Schill, editors, *Proceedings of KI 2006*, volume 4314 of *Lecture Notes in Artificial Intelligence*, pages 419–433. Springer Verlag, 2006.

[69] W. Jamroga. Markov Temporal Logic. Technical Report IfI-07-11, Clausthal University of Technology, 2007.

[70] W. Jamroga. Reducing knowledge operators in the context of model checking. Technical Report IfI-07-09, Clausthal University of Technology, 2007.

[71] W. Jamroga. A temporal logic for Markov chains. In *Proceedings of AAMAS'08*, pages 697–704, 2008.

[72] W. Jamroga. A temporal logic for multi-agent MDP's. In *Proceedings of the AAMAS Workshop on Formal Models and Methods for Multi-Robot Systems*, pages 29–34, 2008.

[73] W. Jamroga and T. Ågotnes. Constructive knowledge: What agents can achieve under incomplete information. Technical Report IfI-05-10, Clausthal University of Technology, 2005.

[74] W. Jamroga and T. Ågotnes. Modular interpreted systems: A preliminary report. Technical Report IfI-06-15, Clausthal University of Technology, 2006.

[75] W. Jamroga and T. Ågotnes. What agents can achieve under incomplete information. In *Proceedings of AAMAS'06*, pages 232–234. ACM Press, 2006.

[76] W. Jamroga and N. Bulling. A framework for reasoning about rational agents. In *Proceedings of AAMAS'07*, pages 592–594, 2007.

[77] W. Jamroga and N. Bulling. A logic for reasoning about rational agents. In F. Sadri and K. Satoh, editors, *Proceedings of CLIMA '07*, volume 5056 of *LNCS*, pages 42–61, 2008.

[78] W. Jamroga and J. Dix. Do agents make model checking explode (computationally)? In M. Pěchouček, P. Petta, and L.Z. Varga, editors, *Proceedings of CEEMAS 2005*, volume 3690 of *Lecture Notes in Computer Science*, pages 398–407. Springer Verlag, 2005.

[79] W. Jamroga and J. Dix. Model checking strategic abilities of agents under incomplete information. In M. Coppo, E. Lodi, and G.M. Pinna, editors, *Proceedings of ICTCS 2005*, volume 3701 of *Lecture Notes in Computer Science*, pages 295–308. Springer Verlag, 2005.

[80] W. Jamroga and J. Dix. Turning game models turn-based for model checking properties of agents. In Katja Verbeeck, Karl Tuyls, Ann Nowé, Bernard Manderick, and Bart Kuijpers, editors, *Proccedings of BNAIC*, pages 143–150, 2005.

[81] W. Jamroga and J. Dix. Model checking ATL_{ir} is indeed Δ_2^P-complete. In *Proceedings of EUMAS'06*, 2006.

[82] W. Jamroga and J. Dix. Model checking abilities of agents: A closer look. *Theory of Computing Systems*, 42(3):366–410, 2008.

[83] W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2–3):185–219, 2004.

[84] W. Jamroga and W. van der Hoek. Strategic ability under uncertainty. Technical Report IfI-05-06, Clausthal University of Technology, 2005.

[85] W. Jamroga, W. van der Hoek, and M. Wooldridge. Intentions and strategies in game-like scenarios. In Carlos Bento, Amílcar Cardoso, and Gaël Dias, editors, *Progress in Artificial Intelligence: Proceedings of EPIA 2005*, volume 3808 of *Lecture Notes in Artificial Intelligence*, pages 512–523. Springer Verlag, 2005.

[86] G. Jonker. Feasible strategies in Alternating-time Temporal Epistemic Logic. Master thesis, University of Utrecht, 2003.

[87] M. Kacprzak and W. Penczek. Unbounded model checking for Alternating-time Temporal Logic. In *Proceedings of AAMAS-04*, 2004.

[88] J. G. Kemeny, L. J. Snell, and A. W. Knapp. *Denumerable Markov Chains*. Van Nostrand, 1966.

[89] D. Koller and N. Megiddo. The complexity of twoperson zero-sum games in extensive form. *Games and Economic Behavior*, 4:528–552, 1992.

[90] B. Konikowska and W. Penczek. Model checking for multi-valued computation tree logic. In M. Fitting and E. Orlowska, editors, *Beyond Two: Theory and Applications of Multiple Valued Logic*, pages 193–210. 2003.

[91] B.P. Kooi. Probabilistic dynamic epistemic logic. *Journal of Logic, Language and Information*, 12(4):381–408, 2003.

[92] O. Kupferman, M.Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, 2000.

[93] P. Lamarre and Y. Shoham. Knowledge, certainty, belief, and conditionalisation (abbreviated version). In *Proceedings of KR'94*, pages 415–424, 1994.

[94] F. Laroussinie. About the expressive power of CTL combinators. *Information Processing Letters*, 54(6):343–345, 1995.

[95] F. Laroussinie, N. Markey, and G. Oreiby. Expressiveness and complexity of ATL. Technical Report LSV-06-03, CNRS & ENS Cachan, France, 2006.

[96] F. Laroussinie, N. Markey, and Ph. Schnoebelen. Model checking CTL+ and FCTL is hard. In *Proceedings of FoSSaCS'01*, volume 2030 of *Lecture Notes in Computer Science*, pages 318–331. Springer, 2001.

[97] A. Lluch-Lafuente and U. Montanari. Quantitative μ-calculus and CTL based on constraint semirings. *Electr. Notes Theor. Comput. Sci.*, 112:37–59, 2005.

[98] K. Lorenz and P. Lorenzen. *Dialogische Logik*. Darmstadt, 1978.

[99] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer-Verlag, 1995.

[100] A. Markov. Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga. *Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete*, 2(15):135–156, 1906.

[101] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford, 1995.

[102] K.L. McMillan. *Symbolic Model Checking: An Approach to the State Explosion Problem*. Kluwer Academic Publishers, 1993.

[103] K.L. McMillan. Applying SAT methods in unbounded symbolic model checking. In *Proceedings of CAV'02*, volume 2404 of *Lecture Notes in Computer Science*, pages 250–264, 2002.

[104] F. Moller and A. Rabinovich. On the expressive power of CTL*. In *Proceedings of LICS'99*, pages 360–369, 1999.

[105] R.C. Moore. A formal theory of knowledge and action. In J. Hobbs and R.C. Moore, editors, *Formal Theories of the Commonsense World*. Ablex Publishing Corp., 1985.

[106] L. Morgenstern. Knowledge and the frame problem. *International Journal of Expert Systems*, 3(4), 1991.

[107] Y. Moses and Y. Shoham. Belief as defeasible knowledge. *Artificial Intelligence*, 64(2):299–321, 1993.

[108] Y. Moses and M. Tennenholz. Artificial social systems. *Computers and AI*, 14(6):533–562, 1995.

[109] J.F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences U.S.A.*, 36:48–49, 1950.

[110] N. J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28(1):71–87, 1986.

[111] M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[112] C.H. Papadimitriou. *Computational Complexity*. Addison Wesley : Reading, 1994.

[113] M. Pauly. Game logic for game theorists. Technical Report INS-R0017, CWI, 2000.

[114] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[115] W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. In *Proceedings of AAMAS'03*, pages 209–216, New York, NY, USA, 2003. ACM Press.

[116] A. Prior. *Past, Present and Future*. Clarendon Press : Oxford, 1967.

[117] W. Quine. Quantifiers and propositional attitudes. *Journal of Philosophy*, 53:177–187, 1956.

[118] F. Raimondi and A. Lomuscio. The complexity of symbolic model checking temporal-epistemic logics. In L. Czaja, editor, *Proceedings of CS&P'05*, 2005.

[119] E.H. Ruspini, J. Lowrance, and T. Strat. Understanding evidential reasoning. *Artificial Intelligence*, 6(3):401–424, 1992.

[120] Tuomas W. Sandholm. Distributed rational decision making. In Gerhard Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 201–258. The MIT Press, Cambridge, MA, USA, 1999.

[121] P. Y. Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2), 2004.

[122] Y. Shoham and M. Tennenholz. On the synthesis of useful social laws for artificial agent societies. In *Proceedings of AAAI-92*, 1992.

[123] I. Ståhl. *Bargaining Theory*. Stockholm School of Economics, Stockholm, 1972.

[124] R. Stalnaker. On the evaluation of solution concepts. *Theory and Decision*, 37(1):49–73, 1994.

[125] R. Stalnaker. Knowledge, belief and counterfactual reasoning in games. *Economics and Philosophy*, 12:133–163, 1996.

[126] K. Su, A. Sattar, G. Governatori, and Q. Chen. A computationally grounded logic of knowledge, belief and certainty. In *Proceedings of AAMAS'05*, pages 149–156. ACM Press, 2005.

[127] A. S. Troelstra. History of constructivism in the twentieth century. Technical Report ML-91-05, University of Amsterdam, 1991.

[128] J. van Benthem. Rational dynamics and epistemic logic in games. In S. Vannucci, editor, *Logic, Game Theory and Social Choice III*, pages 19–23, 2003.

[129] W. van der Hoek. Formal comment on W. Jamroga's paper. Presented at FAMAS'03, 2003.

[130] W. van der Hoek, W. Jamroga, and M. Wooldridge. A logic for strategic reasoning. In *Proceedings of AAMAS'05*, pages 157–164, 2005.

[131] W. van der Hoek, A. Lomuscio, and M. Wooldridge. On the complexity of practical ATL model checking. In P. Stone and G. Weiss, editors, *Proceedings of AAMAS'06*, pages 201–208, 2006.

[132] W. van der Hoek, M. Roberts, and M. Wooldridge. Social laws in alternating time: Effectiveness, feasibility and synthesis. *Synthese*, 156(1):1–19, 2005.

[133] W. van der Hoek and M. Wooldridge. Tractable multiagent planning for epistemic goals. In C. Castelfranchi and W.L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-02)*, pages 1167–1174. ACM Press, New York, 2002.

[134] W. van der Hoek and M. Wooldridge. Cooperation, knowledge and time: Alternating-time Temporal Epistemic Logic and its applications. *Studia Logica*, 75(1):125–157, 2003.

[135] S. van Otterloo and G. Jonker. On Epistemic Temporal Strategic Logic. *Electronic Notes in Theoretical Computer Science*, XX:35–45, 2004. Proceedings of LCMAS'04.

[136] S. van Otterloo and O. Roy. Verification of voting protocols. Working paper, University of Amsterdam, 2005.

[137] S. van Otterloo, W. van der Hoek, and M. Wooldridge. Preferences in game logics. Preliminary version, unpublished manuscript, 2004.

[138] S. van Otterloo, W. van der Hoek, and M. Wooldridge. Preferences in game logics. In *Proceedings of AAMAS-04*, pages 152–159, 2004.

[139] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behaviour*. Princeton University Press: Princeton, NJ, 1944.

[140] D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-time temporal logic with explicit strategies. In Dov Samet, editor, *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK XI)*, pages 269–278, Brussels, Belgium, June 2007. Presses Universitaires de Louvain.

[141] G. Weiss, editor. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press: Cambridge, Mass, 1999.

[142] S. Wlfl. Qualitative action theory: A comparison of the semantics of Alternating-time Temporal Logic and the Kutschera-Belnap approach to agency. In *Proceedings of JELIA'04*, volume 3229 of *Lecture Notes in Artificial Intelligence*, pages 70–81. Springer Verlag, 2004.

[143] M. Wooldridge. *Reasoning about Rational Agents*. MIT Press : Cambridge, Mass, 2000.

[144] M. Wooldridge. *An Introduction to Multi Agent Systems*. John Wiley & Sons, 2002.

[145] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.